Title
Computationally-Mediated Interactions with Traditional Textile Crafts

Author
Grow, April M

Publication Date
2019

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

COMPUTATIONALLY-MEDIATED INTERACTIONS WITH TRADITIONAL TEXTILE CRAFTS

A dissertation submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

April Marie Grow

December 2019

The Dissertation of April Marie Grow is approved:

Professor Noah Wardrip-Fruin, Chair

Professor Michael Mateas

Professor Katherine Isbister

Professor Gillian Smith

Quentin Williams
Acting Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

Abstract

Computationally-Mediated Interactions with Traditional Textile Crafts

by

April Marie Grow

Crafting is a fundamental part of humanity; it gives us purpose and satisfaction, and it produces items that are beautiful and functional. At the same time, our relationship to traditional physical crafts has seen a massive upheaval due to technological advancements. The freedom and support to create textile crafts for pleasure and passion rather than necessity has never been more widespread. Interdisciplinary explorations of computer science and textile crafts have been increasing due to more people being intrinsically motivated to craft, more forms of supportive technology constantly being built, and more interconnected communities growing via the internet. The work in this dissertation begins to coalesce these user-focused and craft-based experiences by presenting crafting as a series of interconnected and nested processes covering the design, manufacture, and experience of textile craft products. Within these processes, I examine how crafters experience physical and digital forms of agency, both high and low, as well as ludic engagement or the lack thereof, as a means of evaluating user perception of technology related to textiles. As exemplar artifacts, I present my own, as well as collaborative, research on design support software, a manufacturing machine appropriated as a game console, and a pair of electronic textile-based fidget toys. This research illustrates how many varied design choices affected audiences' agency and ludic engagement by examining crafters' perceptions and skill levels, as well as the craft domain interpretations by accompanying software and hardware. The interdisciplinary work presented in this dissertation crosses the boundaries

of creativity support software, computational creativity, game studies, human-computer interaction, and crafting communities outside the realm of academia. More importantly, this research begins to explicitly join predominantly feminine craft and masculine technological communities across all ages for the enrichment of all those involved.

To my family. To my friends. To my tenacity.

Acknowledgments

I would like to acknowledge my primary advisors, Noah Wardrip-Fruin and Michael Mateas, for their patience and open-mindedness, especially in allowing me to pursue a craft-based dissertation outside of their areas of expertise.

In addition to my primary advisors, Katherine Isbister and Gillian Smith acted not only as committee members for this dissertation journey, but also as mentors, and they both supplied invaluable personal support and motivation.

I would also like to acknowledge my many research collaborators, co-authors, and lab associates for helping keep me sane and also for helping make our research the best it could be. In particular, Anne Sullivan, Kate Compton, and Melanie Dickinson provided constant encouragement, support, and many pet animals to love throughout this long journey. Ben Samuel and Aaron Reed not only acted as inspiring teaching assistants during my undergraduate career, but also incredible collaborators when I officially joined graduate school. Johnathan Pagnutti and Stella Mazeika provided endless hugs, cuteness, and discussions of the best games to play. Finally, Jacob Garbe deserves acknowledgement for his constant support and impenetrable deadpan delivery to my gullible self.

Chapter 1

Introduction

> Has the pen or pencil dipped so deep in the blood of the human race as the needle? [166]

Crafts are the result of planning, making, building, creating, and/or executing some idea or object, whether it require manual dexterity, artistic skill, job experience, or none of the above.[1] Techniques, terminology, tools, and technology have evolved over the millenia, but the need for personal fulfillment in skilled labor and the need for its end products are as strong as they have ever been [217, 50, 189, 71]. Clothing that we all use every day is an example of textile products, the result of one domain of crafting and the topic of this dissertation. However, it is most likely that the textiles we currently wear and use were manufactured on industrial machines: looms and knitting machines that create yards of fabric much faster than a human could.

Outside of industrial crafting machines, domestic[2] and amateur[3] crafters continue to create their own works on small and individual scales. Technology has been made specifically to target these domestic crafters: digital design tool soft-

[1] As seen in the definitions from the Cambridge[62], Oxford [64], and Merriam-Webster [63] dictionaries.

[2] As opposed to industrial.

[3] As opposed to professional.

ware to create patterns and designs, machine interfaces to allow for machine-human collaboration in the crafting process, and even conductive textile materials that can be used with hobby electronics. The following dissertation aims to examine these three categories of technology and explore the potential for agency and ludic[4] engagement for those who design, make, and experience domestic textile crafts. The following work is interdisciplinary across computer science (CS), human-computer interaction (HCI), game design, and physical computing. By cross-referencing these different research communities, this dissertation shows benefits of interdisciplinary work while suggesting many areas of future academic research.

1.1 Crafts

The definitions of crafts and the related arts have changed over the years (see section 2.1 for more detail). The difficulty of the general concept of crafting as "creation" or "making" is that it is so vague: "...every object must be made in some way, and hence could be considered in some sense to be crafted" [9]. There are endless distinctions in the definition of craft, such as in its domains and subdomains, over the level of skill or unskilled labor, professional or otherwise, and the colloquial "by hand" characteristic, as opposed to using machine support or automated methods (discussed in section 2.1). **Unless otherwise noted, the crafts discussed in this dissertation are distinguished by being: domestic, amatuer and textile crafts.** It is also important to note that all three of these distinctions have historically been associated with women's crafts, to be elaborated on below. A common theme throughout the dissertation is an outreach

[4]Focusing on ludic engagement more accurately describes the projects in this dissertation, as well as making it easier to identify and define than general "engagement."

towards women using textile craft technologies.

1.1.1 Domestic Space

> Records show that, before the invention of the steam engine and the great factory machines that it could run . . . Most of the hours of the women's day, and occasionally of the man's, were spent on textile-related activities. [20]

Due to textile crafts' historical connection to women's private labor in the home, the domesticity of textile crafts has a millennia-long history. The "great factory machines" that Barber describes now mass-produce incredibly affordable textile crafts, and have reduced the necessity of multitasking every moment of the day with spinning, sewing, or knitting. The machines of mass-production work as part of an *industrial* pipeline and are generally[5] not operated by those who craft at a small scale. Another way to distinguish domestic versus industrial space is the "workmanship of risk" versus the "workmanship of certainty" of David Pye [150]. Pye heavily favored the domestic workmanship of risk due to its creative freedom, the marks of individuality to each piece, and its allowance for the possibility of the craft to be worse or (hopefully) better than the intended design. This dissertation primarily focuses on *domestic*[6] tools, techniques, and technology rather than industrial forms of textile crafts. Note that industrial machines appropriated for domestic purposes will be considered as being part of domestic textile crafts.

1.1.2 Amateur Crafters

> We need to recognize the existence of a new super-connected amateur who, informed by a wealth of on- and offline resources (citizen jour-

[5]Small business and home versions of some industrial machines have entered the domestic space (sewing and embroidery in particular).

[6]There are names for domestic amateur crafts that are explained in the background section 2.1.

> nalism, community broadband, online forums, social media), as well as their individual life experiences and expertise, are quietly active as they open up new channels of value and exchange by engaging in alternative craft economies and harnessing assets in often surprising, productive ways. [80]

While *amateur* can indicate skill level for a crafter, that is not necessarily the case. “Amateur” comes from the Latin *amare* (to love), and when used here indicates that the crafter acts in the “spirit of self-gratification” [9]. The amateur is generally not under any financial obligations or deadline restrictions, nor has need to satisfy a patron [103]. The wealth of free information on the internet, as well as a wealth of crafting communities, has further emboldened, informed, and enabled the amateur crafter for passion and self-learning in their craft topic(s) of choice [9, 71].

To give context to what is meant by “amateur crafters,” I give a brief mention to what generally is not amateur. Professional crafts people — those who attend art school for traditional and historical education behind their craft and carry on to be critiqued by “the art world” — are generally not amateur [9]. However, many of these professionals do craft for the love of their work, so the delineation is not as clear as it could be: “The women who developed skills into paid work demonstrate the potentially porous boundaries between amateur and professional making” [80]. The blurred line between pure amateurs and those that devote a lifetime to professional craftsmanship as their primary source of income indicates that the following dissertation offers research contributions towards both communities. However, there will be bias toward amateur crafters in this dissertation, and (as noted above) any reference to crafts and crafters will be to amatuer crafters unless otherwise stated.

1.1.3 Textile Crafts

> On the one hand they respect the tradition of needlework . . . Yet there is also the feeling that by sewing they are practicing a painstaking feminine craft which has low status and strong domestic connotations - Pennina Barnett as quoted in [9].

Textile crafts have a long, worldwide, and inseparable history with domestic crafters and feminine connotations, which has only recently been challenged by subversive means [80, 44, 184]. Many of the projects in this dissertation were created as a means of challenging feminine norms by integrating software and hardware, primarily realms of masculinity, with the femininity of domestic textile crafts. However, the deep gender politics and creative means of protest via *craftivism*[7] are topics for a dissertation, or many, in and of themselves.

Textile crafts have been chosen as the topic for the dissertation as they have a long and rich history of economic and gendered opinions that have changed over time, as well as a very close relationship to technology. *Digital software design tools* help design both industrial and domestic patterns, as well as control a wide variety of physical machines. These machines manufacture textiles of all kinds at different levels of scale, including those in the domestic space. The physical textiles themselves have even been combined with electronics. Textiles cover a wide range of rich crafting domains and topics that match with all sections of this dissertation.

What crafting domains are considered textile crafts? In general, things related to fibres or fabric are considered textiles: plant and animal fibres, man-made materials like plastic and metal made into fibres and thread, and materials made of those products like yarn and fabric. There are broad categories of crafts using these materials, such as *embroidery*, that include specific techniques so common

[7] A more detailed overview of craftivism can be found in the background section 2.1.2.

that they deserve their own distinctions, such as cross stitch, needlepoint, ribbon embroidery, hand embroidery, and machine embroidery. There are blurred lines, where *sewing* generally means garment construction or a relation to fashion, even though quilting and embroidery both necessitate sewing as an activity. The specific textile topics included in this dissertation are (in rough order of appearance): machine embroidery and hand embroidery (3.2), machine knitting and hand knitting (3.3), crochet (3.4.1), quilting (3.4.3), felting (4.1.2), weaving (4.1.3), and macramé (4.1.3), and the broad field of electronic textiles (e-textiles) (chapter 5).

1.2 Crafting Processes

> Thus far I have been employing 'craft' rather loosely, as a word, an idea, and a category. Of course, it can be all these things, but it might be more usefully conceived as a process. [9]

Crafting is a verb, an activity, and as we mentioned above, is impossibly broad, covering all that is making. We must break down the process of crafting into components so that it can be usefully examined. Computational creativity research has been tackling the topic of evaluating the crafting process, and a recent and thorough survey of computational creativity evaluation theories categorize their evaluation techniques using Rhodes' four P taxonomy [153] (Figure 1.1): Person, Process, Product, and Press [106]:

- Person is the human (or non-human agent) who is seen as creative. Person theories study what it is about the agent that makes them creative.
- Process is the set of internal and external actions the agent takes when producing a creative artifact. Process theories study what sort of actions are undertaken when creative work is done.

- Product is an artifact, such as an artwork or a mathematical theorem, which is seen as creative or as having been produced by creativity. Product theories study what it is about the product that makes it worthy of being called creative.

- Press is the surrounding culture which influences people, processes, and products and which judges them as creative or uncreative. Press theories study what it is that leads a culture to view something as creative.

To provide further merit to this taxonomy, example uses of this categorization strategy include Colton's Creative Tripod [37] as press, because it focuses on convincing the audience of creativity; Colton's FACE & IDEA systems [38] as process and press theories respectively; and Jordanous' SPECS criteria [92] as arguably all four theories, as analysed by Bhattacharjya [23].

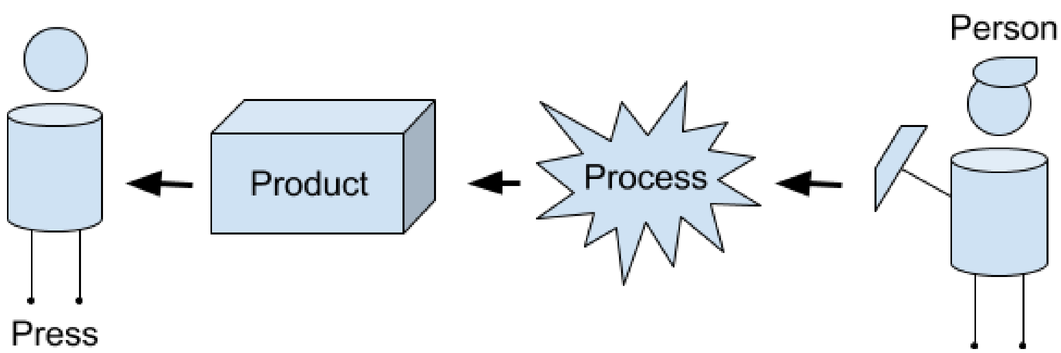


Figure 1.1: Crafting process model, in terms of Rhodes' 4 P's of Creativity. [153]

While this dissertation does include feedback from the Press part of this model, the focus is on the Person, Process, and Product. However, the terminology and roles that have evolved have led to me re-evaluating the terms and roles in this model (Figure 1.2).

As you'll find in many of the projects of this dissertation, this overall crafting

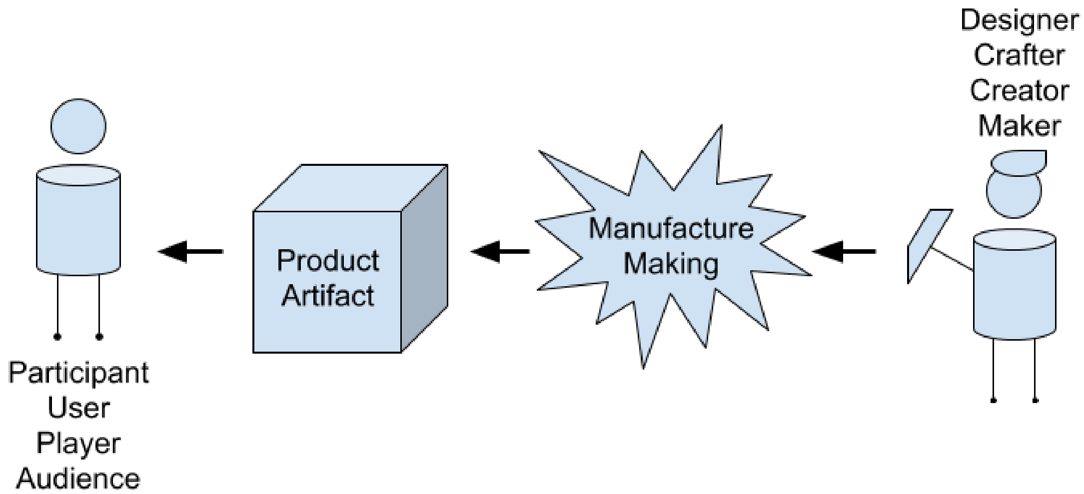


Figure 1.2: Crafting process model, with common modern terminology.

process is not nearly as smooth as this appears. Many designers work in an iterative fashion, and many crafters (or manufacturers) take liberties with their interpretation of patterns. Crafting domains are also often nested: carding and cleaning wool can lead to spinning yarn, spinning can lead to dying the yarn, knitting or crochet can take the dyed yarn and turn it into fabric. A more accurate model might look like Figure 1.3.

Nested processes can be part of the same manufacturing processes, or the product of each stage can be part of the input to another process, possibly separated by dozens of miles or years. For this nested reason, the "crafting process" can mean the whole model from start to finish, one stage of the model, or it can be part of the big-P "Process" that Rhodes uses.[8] In this dissertation, crafting processes in plural refers to the whole model, while if a specific singular crafting process is used, it should be clear with context or modifiers which part of the model is being addressed.

[8]One motivation for using updated terminology for this model is the overlapping uses of the term "process."

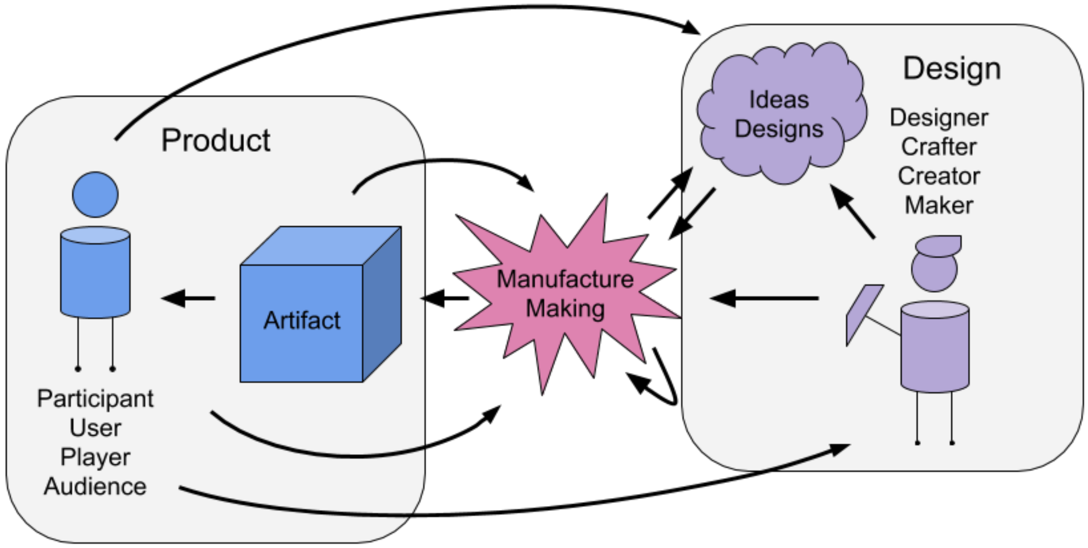


Figure 1.3: More accurate *overall* crafting processes model that represents the iteration found at all levels of crafting processes. These three parts: Design, Manufacture, and Product match the three main chapters of this dissertation.

1.3 Agency and Ludic Engagement

Not all textile crafts are inherently interactive. Most textile pieces displayed in galleries and exhibitions are bestowed with a *Do not touch* sign or displayed out of reach, robbing the audience of any hope of appreciation outside of their sense of sight [128]. However, the craft processes described in the last section, and especially in Figure 1.3, show a very interactive and iterative set of probably nested processes. Not only are processes interactive at a high-level, but the tools afforded by supportive design software, machines, and the communities surrounding the crafting domain and its tools all offer further means of interaction with the textile medium.

Within these interactions, the crafter is constantly making choices: high-level design decisions and low-level moment-to-moment actions. The available options to these choices relate to a sense of agency, and the actual interaction may be characterized as ludic engagement. This dissertation explores the potential for *agency* and *ludic engagement* for all roles — designer, crafter and participant — in all three of the crafting processes: design, manufacture and product. Ludic engagement is chosen, rather than general engagement, as it more closely fits with my background and research focus. This dissertation crosses many academic communities, including game studies and experimental game design, and ludic engagement is a more accurate description of the engagement I wish to examine in crafters. It should also be made clear that the topics of agency, specifically digital agency, are being appropriated from game studies research to apply to digital design tools as a similar means of interactive software.

Agency is a complicated topic that stretches across psychology, philosophy, and sociology. Along with the rise of computing, agency has been explored in digital spaces, where the capacity to act is not restricted by the physical world. The work in this dissertation crosses the digital and physical boundary; crafts that have been an entirely physical affair have been translated to digital forms, most commonly to design tools that provide a means of crafting or designing patterns entirely within software. Having a digital representation of crafts allows a crafter without the motor functions to execute a pattern (physical agency) to design patterns via software interfaces (digital agency), and a crafter that believes they are incapable of designing a pattern using supportive design software (digital agency) may follow a pattern and successfully make a physical product (physical agency). The following sections introduce the two concepts of agency separately and describe the conditions for ludic engagement.

1.3.1 Physical Agency

Physical agency is our perception of our freedom of choice and our sense of control, whether it exists (a *judgement* of agency) or potentially not (a *feeling* of agency), and how the intention of our choice manifests [130]. Physical agency applies to crafters when following and actuating a pattern: do they have the means to execute the pattern to their satisfaction? Crafters without a feeling of agency may have a physical impairment that has convinced them of their inability to craft, or they may doubt their own capability to execute a pattern. Indeed, physical impairments are often reinforced by a judgement of low agency. Crafters that attempt a pattern may also discover they have a judgement of low agency, as demonstrated via the moniker "pinterest fails" (Figure 1.4).

Figure 1.4: Pinterest [148] is image and video visualization, aggregation, and sharing software primarily aimed at crafters. Pinterest fails occur when a crafter attempts to follow a tutorial or to recreate an image likely found on Pinterest with disastrous results. [54]

Part of a crafter's sense of agency is in knowing their capabilities and in picking patterns that are appropriate for them. A beginner knitter should not attempt a sweater, or even a sock, unless they are willing to learn many new skills along the way such as knitting in the round, executing short rows, shaping, and fitting designs. That is not to say that a knitter has never successfully made a sweater for their first project, but it takes a certain level of patience, persistence, and willingness to learn, or possibly low expectations for a polished result, to feel a sense of agency in the face of high difficulty. A wide selection of patterns with different techniques will not only entice and inspire crafters with possibilities, but also allow the crafter to better pick an appropriate pattern. Increased availability to additional tools, techniques, and materials also increases the range of feasible patterns. The more of these appropriate pattern possibilities that the user surveys, the more empowered and knowledgeable they are to make a choice that fits their capabilities and their desires. Up to a point of over-saturation and choice paralysis,[9] the increase in options endows the optimistic crafter with a greater sense of agency. A subset of crafters, often the ones above that do not feel a sense of agency to begin with, may be intimidated into inaction if they cannot find any options that fit their capabilities:

> The next, and arguably most important, thing to remember is that we're not all Pinterest stars. No matter how much money, time and dedication that you put into a craft, some of us will never have the knack that it takes to start our own Etsy[10] store. [51]

Pattern designers can also increase a crafter's sense of agency with a well-crafted and well-presented pattern. A high-quality pattern generally includes many in-progress photos, charts, and/or diagrams, photos of the end product, a

[9] Too many choices can lead to an inability to judge which option is better (analysis paralysis), which leads to no choice being made (choice paralysis). The precise threshold of choice paralysis is unknowable, as it changes constantly for different kinds of choices and the chooser's disposition.

[10] Etsy [66] is an online host and aggregator for crafters to setup digital storefronts to sell craft materials and final products directly to consumers.

full materials list, definitions and tutorials (or links to them) for all techniques involved, and a set of detailed instructions that have been verified by multiple crafters that are not the original designer [15]. A pattern with these properties will inform crafters about the pattern's difficulty and allow them to more accurately assess how well-equipped they are for executing the pattern.

1.3.2 Digital Agency

In a digital environment, a designer of a supportive software, whether for designing human patterns or controlling machines, may fully control the realm of possibilities available to the user. Whether or not the user understands the range of possibilities, the range of their choices within the possibility space, or the ramifications of those choices, is unknown and dependent on the individual user. Digital design spaces luckily offer the safest and least costly crafting environment to explore their craft possibility spaces. My chosen basis for agency for digital spaces is adapted from agency in video games, as it is closely related to ludic engagement and the many games projects in this dissertation. However, my interpretation, described below, treats the crafter's skill level as a major consideration of their agency.

In video games, designers engineer agency, balancing what agency is suggested to the player via the world and its domain (formal affordances), and what choices are specifically programmed into the game as possibilities (material affordances)[11] [212]. These affordances are seen from a idealistic perspective, where every audience member perceives and interprets the same formal affordances in the same way and will likely experience the same material affordances. To avoid confusion, I will rename and expand expand upon these affordances in order to adapt them to

[11] Also referred to as constraints.

a more user-centric approach that acknowledges that different users feel different levels of agency based on their unique perspectives.

- I define *expected affordances* as the user's expectations of their software's domain and how that maps onto relevant software tools. A user has expectations on what content should belong to their domain and what features they should see in their software. This is my addition to the digital agency found in [212].

- I define *presented affordances* to be what the software design tool suggests should be possible. This is the equivalent to formal affordances described above. For example, in all domains of software design tools[12], a robust save, load, undo, and redo system is an expected affordance. By declaring a piece of software a design support tool, the tool suggests these affordances be presented to the user.

- I define *perceived affordances* to be what the user perceives the software design tool as being *actually* capbable of. Continuing the previous example, if a software design tool does not have any button menu, history stack, or other visual indication of saving, loading, undoing, and redoing, then the software is perceived to not have those features. Those features may still exist, perhaps only as keyboard shortcuts, but an individual's expected affordances of having a visual interface, like all other design support software, are not satisfied if they cannot perceive those features. This is roughly equivalent to material affordances from above.

The fullfillment of a user's expected affordances via presented and perceived affordances is necessary for high digital agency. If a user has

[12] Alternatively, creativity support tools, see sections 2.2.1 and 3.1.1.

low expected affordances, and their software has the appropriate (few) presented affordances that the user is capable of perceiving and using, then there is a balance of these three affordances and a sense of high agency. If a user has high expected affordances, then their software must present the (many) appropriate software features in a familiar way that the user may perceive in order for them to have a sense of high agency. In either of these cases, if the features the user expects are not there in the software, or the user cannot find or use them to their expectations, then there is an imbalance of these affordances and a result of low digital agency.

As just previously mentioned, standard features of design support software, such as saving and loading designs and undoing and redoing edits to the design, foster safety and trust with the designer in using the tool [175]. These baseline affordances are *expected* and *presented* by any design software. What the user further expects to see as features in their design software depends on the craft domain and their familiarity with it. For example, a cross-stitch chart-making software needs an advanced grid that allows splitting a grid square into quarters for partial stitches, as well as outlines in between and across stitches. A chart for knitting may have special symbols for cables that span a large block of grid squares. A color chart for bead weaving designs may want a brick stitch layout where every other row is offset by a half of a grid square. All three of these domains could be represented using the same grid, but their charts each have additional design features that need additional support (Figure 1.5). These domains illustrate different expected affordances that suggest needed presented and (hopefully) perceived affordances within their respective design tools.

Physical agency in a craft domain influences this appropriation of digital agency. A user's perception of expected affordances (from the craft domain) is based on their personal experience and their design goals in using a particular

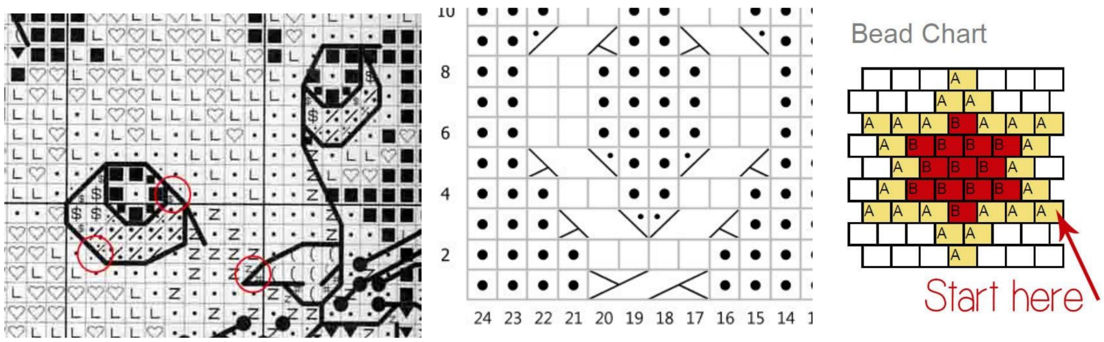


Figure 1.5: Example charts for (right to left): cross-stitch [198], knitting [16], and bead weaving [48]. The cross-stitch chart highlights partial stitches in the pattern; the knitting chart illustrates cable crossings, and the bead chart has its rows offset for brick stitch.

design software. An experienced machine embroiderer knows the capabilities of their machine and has high expectations of their design software to offer control of those features. In order to have high digital agency, the software the experienced machine embroiderer desires to use should present a wide variety of fine tools (many presented affordances) with which the user can operate their machine (perceived affordances). Likewise, an inexperienced machine embroiderer does not understand the breadth of the medium (low expected affordances) and would need to be presented with a simpler interface (low perceived affordances) and given less chances for operator error (few presented affordances). In both cases, if the high or low expected affordances match the high or low presented and perceived affordances, respectively, the user experiences high digital agency.

In summary, the agency experienced by the user is dependent on their experience and design goals, how well their chosen design software matches those goals, and the user's perception of the design software. This dissertation has a bias towards novice users/crafters (lower expected affordances) due to the likelihood of them being amateurs; if user perception is not specified, it is likely that they need lower presented and perceived affordances to match their low expected affordances in order to experience high digital agency.

1.3.3 Ludic Engagement

Ludic engagement is a playful means of interaction, or "designs that privilege pleasure over function" [131]. This engagement is "motivated by curiosity, exploration, play, and aesthetics rather than externally defined tasks" [146]. As we outlined previously (in section 1.1.2), the idealistic amateur operates without material constraints,[13] without the creative constraints of external validation and professional necessity, and with access to all manner of materials, tutorials, and supportive communities online. Self-reports by crafters cite the same motivations as ludic engagement as both their motivations to craft and their experience of the crafting process [214, 184, 110]. In particular, "to have fun" is one of the main motivations[14] to craft in general [217]. Many existing examples of professional design software fall under productivity-focused forms of interaction rather than ludic engagement. Further exploration of design software designed to promote productivity and creativity is found in sections 2.2.1 and 3.1.1.

1.3.4 A Summary of Challenges

For the ideal modern domestic amateur crafter described above, their flexibility within and between crafting domains is very high. The power of this theoretical crafter's creative potential and their limitless access to the means to actuate that potential result in a strong sense of physical agency, and their activities should align perfectly with ludic engagement. Realistically, individuals do not have infi-

[13] Not to say that amateur crafters have infinite resources, but their income is not necessarily related to their crafting. Due to its nature as a hobby, a theoretical ideal crafter would be to not be constrained by material concerns. However, in reality, monetary, supply, and space concerns are major blockades for agency and engagement.

[14] More details on what motivates people to craft can be found in section 2.1.3.

nite time and resources to take advantage of this limitless freedom, and thus their sense of physical agency is lower. Many people also do not consider having more options as a positive influence to their confidence and physical agency, and they would likely rather operate under creative constraints [140, 181].

These instances of low feelings of physical agency, as well as imbalanced expected, presented, and perceived digital affordances, are contributors to a low sense of agency, and are most likely a result of the following factors: inaccurate personal assessment[15], choice paralysis, low quality of available patterns[16], lack of access to resources[17], poorly engineered design software, and moment-to-moment feelings of frustration or boredom[18].

A user with a high sense of agency, where the user feels in control of a fully supported range of choices, may interfere with ludic engagement. Where there is full confidence and control, there may be a lack of surprise and serendipity that often contribute to ludic engagement. This trade-off occurs depending on the user's crafting goals. Those crafters with a very specific goal in mind [121, 22], and/or those who are productivity-focused (see section 3.1.1), are likely to have a serious or practical purpose that inhibits the freedom of play and improvisation. Thus, not only does the design of the digital or physical experience guide ludic engagement, but the specific participant's mindset and goals influence ludic engagement, for better or worse.

Breaking down the impediments to crafting agency and engagement by the different crafting processes will help us address different agency and ludic engagement barriers that plague crafters at different stages in their crafting journey.

[15]Inaccurate assessment includes both over- and under-estimating one's capabilities, which may or may not be informed by previous failures.

[16]More information in section 4.4.2.

[17]Material resources can include money, supplies, a crafting space, and time.

[18]Feelings of frustration and/or boredom indicate a crafter has fallen out of the flow state, expanded on in section 2.1.4.

1.4 Research Contributions

The research on domestic amateur textile crafters tends to paint a picture of an ideal model: a crafter full of freedom and capabilities, able to find, learn, and make whatever their heart desires. In some respects, this is true. However, the crafting model would not be as nested or iterative if crafting had no challenges. This dissertation explores how modern technology can support crafters and provide new avenues of exploration and inspiration for their work.

1.4.1 The Crafting Model

The overarching structure of the dissertation reaffirms the three main stages of the revised crafting model of Figure 1.3: design, manufacture, and product. While the original model was specifically aimed at *creativity*, I argue that this adapted model better fits the topic of actual crafting. Firstly, design and planning is elevated to its own top-level process as part of the crafter, as *patterns*[19] made in this process play a crucial role in what and how crafters craft, as patterns designed, adapted, or simply followed in their crafting processes. Secondly, Rhodes' "Process" has been renamed to manufacture to relieve overloading of the term process [153]. Thirdly, the audience experience of the product has been folded into the end artifact, the goal of the crafting processes, as they give context and meaning to each other. While, as previously stated, these crafting processes are nested and iterative, and thus make the boundaries less clear, the work presented in each following chapter should demonstrate the practical usefulness of this revised model.

[19] A major topic of chapter 3 and section 4.4.2.

1.4.2 Research Questions

How can computationally-mediated interactions with physical and digital representations of traditional textile crafts affect the physical agency, digital agency, and ludic engagement of designers, manufacturers, and experiencers of textile crafts?

Designers, manufacturers, and experiencers of modern textile crafts, each augmented with different contemporary software and hardware technology, have different relationships with their medium. These different roles and different crafting skills lead to different perceptions of agency and ludic engagement, even by the same crafter. To address the needs of these different processes, this primary research question is broken down by each process in the model: design, manufacture, and product. It is important to note, however, that the interconnected and nested crafting processes leads to the crossing of boundaries between the roles of designers, manufacturers, and experiencers. The roles highlighted in the following sub-questions are the primary focus of those sections, but considerations for the other roles are also acknowledged.

How can *digital* representations of textile crafts affect *designers'* agency and ludic engagement?

Technological interventions in the (pattern[20]) design part of the textile crafting process come in the form of software-based design tools that offer ideation support. These design tools target a wide range of textile domains, and some of these tools are built specifically to communicate with hardware, such as knitting, embroidery, and quilting machines. Through examination of creativity support tools, I will present an overview of the range of design tools across a range of crafter skill

[20]To more clearly separate the design and manufacture processes, the output of the design process is generally a pattern that the manufacturing process then uses as input. Explicitly designing a pattern is not always a part of a crafter's processes.

levels and design experiences. The core of the chapter will be examining two different design tool projects, one for blackwork embroidery and one for knitting machines, which expose the necessity of creating domain-specific languages for the appropriate textile craft domain that help abstract distracting or difficult elements from the pattern design process. I will elucidate how to design and scope these domain-specific languages, as well as point out the most salient design tool software elements for targeting domestic amateur crafters specifically using elements of ludic engagement that increase their digital agency. This research will offer insights into creating successful pattern design software for specific crafting domains and techniques that do not yet exist or that need refinement.

How can computationally-mediated textile craft *manufacture* affect crafters' agency and ludic engagement?

Once the crafter has a pattern they want to follow, a pattern they want to adapt for their own use, or even just an idea of what they want to make, the creation of the artifact begins! But where is the computationally-mediated element, other than some crafts using a machine as a tool to speed up their work, such as a sewing machine? I will present research that demonstrates an innovative approach to craft manufacture that offers a new form of crafter agency: a collaborative process where humans interact with software and hardware *during* the manufacturing process, where the software or hardware participates in an *active* manner. An active software or hardware participant has some onboard software that interprets user input — a design or other action — and reacts with input rather than only mechanically following orders, in a meaningful way that affects the crafted product. An old sewing machine, such as the example previously mentioned, makes no active choices other than to strictly obey manual input by the user. A modern sewing machine that refuses to sew after the user triggers a safety mechanism, such as a thread break, or an embroidery machine that interprets a user's pattern, are examples of active participants that act *outside of* (or explicitly stop) the manufacturing process.

Many of the projects discussed in the manufacture chapter are games played on or with the software or hardware that invite ludic engagement. During the process of play, a physical craft is made using inputs or guidance from both the human crafter/player and the software or machine. The primary research project I present is a game played on a quilting or embroidery machine, where the machine enforces the game rules and sews the result of game choices that human players make. At the end of the game, a physical artifact of their game trace, along with the score achieved, is sewn on a piece of fabric. I will go into further detail on how the presented projects are considered active participants, how the human and non-human participants interpret the same crafting patterns, how the playful and interactive approach to these projects were the key to their ludic engagement, and how limitations shaped these projects and also made them possible. This research will offer insights into understanding machine limitations and affordances for both designing and participating in these collaborative manufacturing craft processes.

How can the integration of computation with physical textile craft *products* affect the agency and engagement of those who craft and experience textile crafts?

Integrating computation with the product of textile crafts is its own domain of crafting: electronic textiles, or e-textiles. The projects discussed in this chapter use e-textiles as a means of enabling new forms of education, interaction, and play in physical spaces. The affordances of the e-textiles software are exploited in an effort to bring people together physically, as well as to be present in the moment. The primary research project I present is a pair of fidget toy prototypes that make use of a wide range of e-textile techniques in order to track and better understand the fidget behaviors of children. The toys are designed to be as evocative and engaging as possible in order to encourage all kinds of touch-based activities. I will present how amateur textile crafters tackle the difficulties of electrical engineering in making e-textiles, provide an overview of e-textile affordances and design challenges, and examine how different in kind the ludic engagement is in e-textiles compared to the artifacts made in the previous section on mediated manufacture. This research offers an overview of specific design challenges related to e-textiles, how to overcome or mitigate those challenges using the affordances of the physical medium, and a reflection on effective uses of e-textiles.

1.5 Dissertation Outline

Directly following this introduction is a background chapter that provides further information on the context and topics introduced above: crafting, creativity, and the communities that contribute to and are affected by this dissertation.

The core body of the dissertation is split into three overarching chapters that correspond with each of the three crafting phases: design, manufacture, and product. The research question that each chapter addresses, as well as an overview of the content and contributions presented in each chapter, are described above.

Finally, the conclusion reaffirms my research, its contributions, and their impact regarding the presented research questions and their relevance to different research communities. At the end of the conclusion is a brief description of future in-progress research: a co-operative design tool for embroidery machines that explicitly employs the agency and ludic engagement lessons presented in this dissertation. A call to action for further interdisciplinary work completes the dissertation.

Chapter 2

Background

The deep and broad history of creativity, crafts, industrial technology, and computing all mix in this dissertation. It is beyond the scope to fully cover all these topics, even just those relevant to domestic amateur textile crafts. This chapter will cover a brief historical overview of content not covered elsewhere in the dissertation, in so far as they motivate and influence contemporary domestic amateur textile crafts.

2.1 The Evolution of Crafts

This section (The Evolution of Crafts) is adapted from Crafting in Games*, a collaborative journal article with accompanying authors Melanie Dickinson, Johnathan Pagnutti, Noah Wardrip-Fruin and Michael Mateas (for full citation see [77]). Part of my contribution to that work was the history of crafts section that is adapted and expanded upon here.*

Paul Greenhalgh separates the history and characteristics of craft history into three threads: decorative art, the vernacular, and the politics of work [65]. These attributes cover the split between art and craft, public opinion of handmade goods over time, and the connection of crafts to mercantilism and the economy, all of which have affected contemporary textile crafts. In particular, this brief history illustrates the foundation of the gender biases found between art and craft.

2.1.1 Decorative vs. High Art

In the West, roughly since the European Renaissance when Academies were being created, the five fine arts of painting, sculpture, architecture, music, and poetry became privileged over other artistic professions [65]. These privileged arts were predominantly masculine and were further separated from women due to their relationship to higher education. Other arts, which Greenhalgh calls decorative arts, were considered by some to be too functional or made of too-cheap materials for the designation of fine art [65]. These disenfranchised crafts were clearly either for the poor and/or feminine populations and included nearly all forms of textile crafts.

2.1.2 The Vernacular and Craftivism

After the privileging of high art, the vernacular surrounding the decorative arts of rural pre-industrial country craftsmen was unfavorable until the late 19th century, when it took on the perception of being “unpolluted” and “authentic” when compared to mass-produced goods [65]. In general, the 19th century recognized finely made items by men to be more highly elevated, while most feminine textile crafts remained hidden from public attention due to their ubiquity in the home. The common opinion of handmade items has oscillated between these favorable and unfavorable views for the last century. In *Handmade Nation*, Andrew Wagner speaks of the 60’s and 70’s “hippy counterculture,” how 80’s and 90’s artists distanced themselves from hippies by focusing on galleries and museums, and today we have another DIY resurgence [110]. Debbie Stoller in *Stitch 'n Bitch* traces a more nuanced history of knitting in particular: how each generation either snubs the "silly domestic work" or takes up the needles through necessity or pride, such as in times of economic hardship or war [189]. Some alternative terms I have heard refer to this DIY resurgence include maker practice or maker movement, hacker culture, and appropriation of handicrafts [194].

Craftivists, practitioners of a specific subculture of the maker movement, create crafts as a means of political self-expression, such as graffiting with yarn, or "yarn bombing," in the effort to reclaim cold public spaces [132, 120]. Many of the projects presented in this dissertation could be considered transgressive or subversive crafts. The practice of taking "Grandma's crafts" — traditionally feminine crafts, such as scrapbooking, knitting, sewing, and other textile crafts, used to convey "deceivingly harmless" messages reinforcing gender, religious, political, or familial identity or ideology — and inverting, twisting, or otherwise subverting their usual meaning is known as subversive crafting [217]. Figure 2.1 shows a contemporary cross-stitch pattern available [25] on Etsy [66] of a subversive design.

Many sources, including those cited previously as well as David Gauntlett's book *Making is Connecting*, speak to the current renaissance of craft knowledge and products spread throughout the internet [71]. Popular websites such as Ravelry [152], Craftster [49], and Instructables [18] encourage users to share their own patterns; and commerce websites like Etsy [66] or Handmade by Amazon [14] exist for users to sell their handmade goods and materials. Future projects in this dissertation leverage Ravelry as a community for sharing knitting experiments that require or encourage distributed creation. Craftster's message forums for seeking help on all forms of textile crafts are incredibly active. Communities like these

Figure 2.1: A cross stitch of a subversive message championing both sexual consent and political unrest accented with feminine floral motifs. Designed and sold by BitchinStitchesByLiz [25]

are the best places to survey the challenges crafters report having during their crafting processes.

2.1.3 Craft and Creativity

The Industrial Revolution's toll on an individual's creative freedom was being felt widely enough to seed an international shift in values, the Arts and Crafts movement, across Europe, in other nations in the British empire, in North America, and later in Japan [65]. This movement defined and popularized the concept of "craft" as this dissertation refers to it: as a creative endeavor alongside and sometimes including the fine arts [65].

What is creativity, then, as a necessary component of a creative endeavor? In general, many researchers of human and/or computer creativity agree that the "standard definition" is some combination of originality (or novelty, or innovation) and effectiveness (or appropriateness, or usefulness, or value) [161].

Related to the evaluation of crafts is motivation: why do people craft? Winge and Stalp did an ethnographic study of 44 crafters, primarily subversive ones, and some of their reasons were: for necessity (or lack thereof), to develop skills (learn), to relax, to have fun, to create a supportive space, and to develop identity [217]). Many find it "therapeutic" to quietly reflect, or think about nothing at all, and to watch a product slowly unfold [217]. To summarize, some craft to make a specific object or outcome, and others for the process of making regardless of the outcome. While the value of a specific object may or may not be worth the cost of admission to create it, the process is often the more interesting motivation.

2.1.4 Flow

The flow state, coined by Mihály Csíkszentmihályi [52], is a psychological concept about a person's mental state while being engaged that has been applied to many fields including creativity [53], creativity support software [176], and games studies [21]. Colloquially known as being "in the zone," this mental state is a balance between boredom and frustration that fully engages a person's senses to the point where they likely aren't even aware of the passage of time [129].

Nakamura and Csíkszentmihályi explored six factors that make up the flow state, two of which explicitly support the argument of this dissertation: "[a] sense of personal control or agency over the situation or activity" and the "[e]xperience of the activity as intrinsically rewarding, also referred to as autotelic experience" [134]. Not only do these two cases of agency and autotelic experiences line up with my adaptations of agency and ludic engagement, but they also reflect the self-reported reasons for crafting found in [217]. It should also not be taken as a set-back that only these two factors so closely match the presented definitions, as the flow state is not the only reported reason for crafting. Figure 2.2 shows a diagram of flow and related mental states regarding the balance of boredom and frustration (the balance of difficulty and skill, also known as challenge and ability). In this figure, we can see that flow, control, relaxation and arguably arousal are all states that crafters report as reasons why they craft.

Furthermore, Schaffer proposed seven conditions for entering the flow state, many of which align with agency and ludic engagement as well [165]. Six[1] of the seven conditions can be summed up as supporting the crafter's confidence in themselves (their abilities) and their chosen task (their challenges):

- Knowing what to do
- Knowing how to do it
- Knowing how well you are doing
- Knowing where to go (if navigation is involved)

[1] The missing seventh condition is "Freedom from distraction" which is often not necessarily in control of the crafter nor necessarily a component of their mental state.

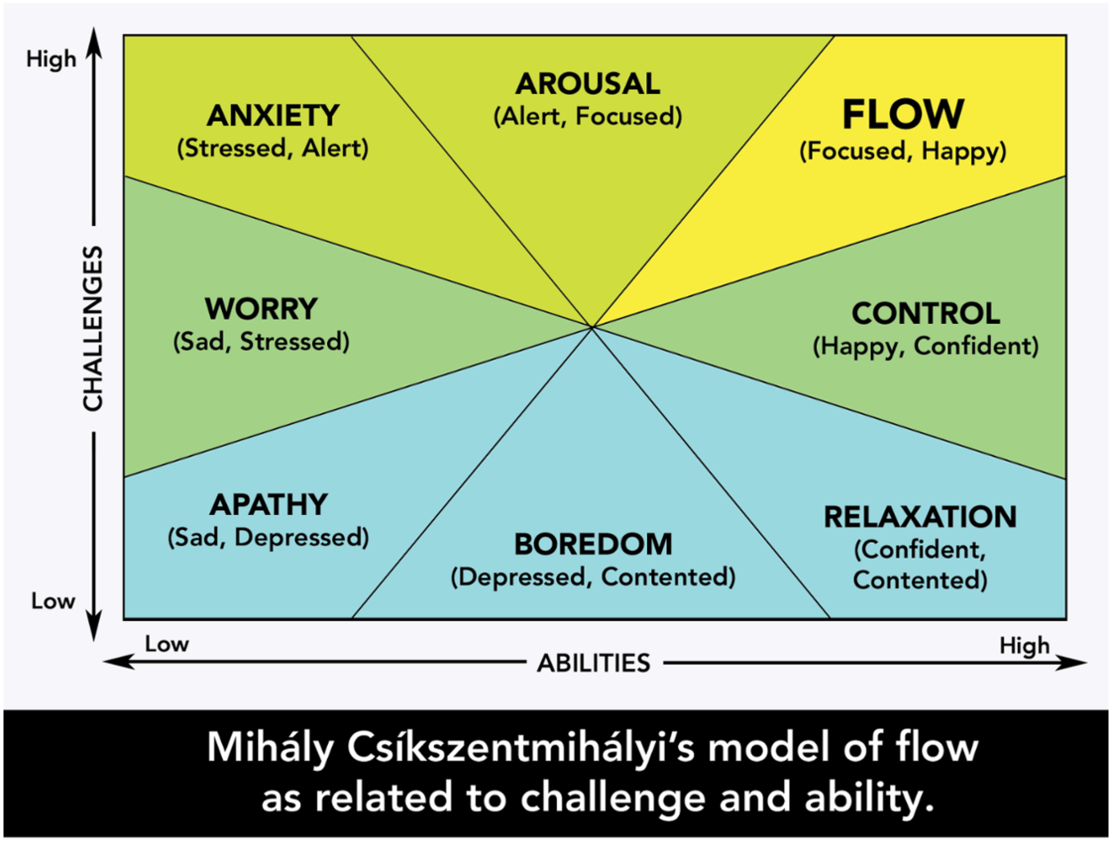


Figure 2.2: A diagram of flow and related mental states regarding the spectrum of challenge vs ability. The right-hand states of flow, control, relaxation, and arguably arousal match descriptions of why crafters craft. Conversely, the left-hand states of anxiety, worry, apathy, and boredom would be reasons why people *don't* craft (as seen in the Introduction section 1.3.4).

- High perceived challenges

- High perceived skills

In particular, the first four — knowing aspects of the crafting processes — are direct representations of feelings of (physical) agency and high expected affordances (in the case of digital agency). Whether or not the crafter can do their task, they are able to make informed choices about the state of their crafting processes. The last two are representations of the crafter's opinion of themselves in relation to their chosen task, which further influence their feelings of agency.

Again, not hitting all of these flow conditions is not necessarily a sign of failure in terms of agency and ludic engagement either, as not every crafter is aiming explicitly for the flow state.

In general, these positive states of flow, control, relaxation, and arousal describe the crafter's motivations of relaxation and fun, as well as being therapeutic. By definition, a lack of boredom and/or frustration that obstruct agency and engagement during crafting processes is indicative of the flow state. Partial conditions and factors of flow are suitable for agency and ludic engagement as well, as crafters have a wide range of motivations in crafting

2.2 Computer Science and Creativity

Understanding "creativity" is not a solved problem, but we have a strong foundation thanks to the flow state's applications toward creativity seen in the previous section. This section provides a brief introduction to research regarding computer science and creativity, which can be summarized with two questions:

1. (How) Can computers support, aid, increase, or inspire creativity in humans?
2. (How) Can computers be creative in their own right or with the help of humans?

When the computer works with the user to contribute to a final product, specifically when it performs some manner of change in the object beyond that of a mere tool, the terms *co-creativity* and *mixed initiative* have been used to refer to the phenomenon [138]. The academic community of computational creativity attempts to directly address and explore the instances of these types of software, their ramifications on human philosophy, and how to approach evaluating them.

Taken one step further, when the focus of the software or algorithm is on generating the content rather than the more cooperative tool format or front-end, the realm of publications shift toward the *procedural content generation* (PCG) community.

2.2.1 Creativity Support

Design support software, also known as authoring tools and the main topic of discussion in chapter 3, are the core of creativity support tools. This brief section will cover a broader background that applies to the whole of the dissertation: creativity support. Crafts are intrinsically tied to creativity, and it is useful to examine how computer scientists support creativity in software and hardware, in contrast to psychology (as in flow).

There are many different approaches to what it means to support creativity, and the pieces of software that support or execute these methods all 'count' as creativity support tools:

> Structuralists believe people can be creative if they follow an orderly method... Inspirationalists argue that breaking away from familiar structures elicits creative solutions... [and] Situationalists recognize that creative work is social. [175]

A comprehensive summary of the purpose (design goals), design principles to accomplish their purpose, and evaluation techniques for creativity support tools was collected during the NSF-Sponsored workshop Creativity Support Tools [176]. The design principles for creativity support tools should at least partially[2] apply to the design tools and other projects presented in this dissertation [176][3]:

[2]One software tool, especially those at a small scale or for a very narrow domain, is not expected to accomplish all the varied tasks listed as the breadth of principles for creativity support tools.

[3]Ben Shneiderman echoes this list in subsequent publications such as [175].

1. Support exploration.
2. Low threshold, high ceiling, and wide walls.
3. Support many paths and many styles.
4. Support collaboration.
5. Support open interchange.
6. Make it as simple as possible — and maybe even simpler.
7. Choose black boxes carefully.
8. Invent things that you would want to use yourself.
9. Balance user suggestions with observation and participatory processes.
10. Iterate, iterate — then iterate again.
11. Design for designers.
12. Evaluate your tools.

A strong theme in this list is explicitly providing support for amateurs: exploration of presumably unknown areas of the domain (1), low thresholds to entry (2), and making it as simple as possible to interact with the artifact[4] (6). The ease of use of software like this directly relates to agency; low presented affordances that are easily perceived increases the likelihood that users are able to take advantage of them in order to meet their expected affordances. Exploration of existing areas of the domain, in rapid and reversible ways, provides a safe and comfortable space for ludic engagement with the domain's possibility space [175].

[4]Can refer to any of the research projects discussed or cited in the dissertation.

Within HCI, the sub-communities of creativity support tools and computational creativity have identified and attempted to wrangle with the issue of measuring productivity, as well as issues relating to creativity as a whole.

2.2.2 Computational Creativity and Mixed-Initiative Co-Creativity

Our most intimate and well-explored perspective on creativity is that of ourselves as humans. Computational creativity aims to understand creativity as it would be applied to and potentially expressed by machines. The computational creativity community directly acknowledges, confronts, and integrates the multidisciplinary history of human creativity directly with computers. Some systems involve full cognitive science models or psychological theories of mind, while others focus on including explicit computational modules that address aspects of creative agency such as contemplation, intention, and reflection on the system's own output.

The conceit of AI expressing and trading initiative with the human user was envisioned as "man-computer symbiosis" in 1960 within the realm of dialog agents [113]. It was claimed to have been accomplished by 1970 when Carbonell stated that their system "can generate text, questions, and corresponding answers" and, in combination with the student's questions and answers in text, form a two-way and mixed-initiative dialog [32]. The mixed-initiative concept has expanded into other contexts of AI and computation, although the extent to which the computer actually takes initiative varies widely. For example, the one major differentiation between creativity support tools and computational creativity is that creativity support tools do not focus on — and often minimize — the role of the computer's contribution (other than to enhance the user's workflow). Afterall, the best interface is no interface [105]. On the other end of the spectrum, the procedural content generation community often puts as much of the burden of the crafting process into the hands of algorithms and software as possible while distancing from the concept of creativity (more on that in the next section).

Suppose a computer takes the initiative in some form and contributes to making some product. Suppose that, by some human or computational measure, that product is deemed "creative." In humans, we would share attribution between creators based on their level of involvement in the creative process. However, many humans are wary of freely granting co-creative ownership or responsibility with a machine. Software that is labeled by its authors as "co-creative" asserts that they deem their software as being an active creative agency. Their software should be granted some amount of creative attribution alongside the human(s) involved, whether it be a human as algorithm-designer, human as software-builder, or human as computational creativity tool user. Evaluation techniques of this kind of software should include some argument toward convincing the broader community that the creative tasks were indeed shared between the human(s) and the computer.

2.2.3 Procedural Content Generation and Generative Design

Procedural content generation (PCG) is a field of study that examines the mostly autonomous creation of digital products by computers [197]. PCG most commonly refers to game assets ("levels, maps, game rules, textures, stories, items, quests, music, weapons, vehicles, characters, etc.") [197], while others have been expanding the model to include any autonomous creation made by computers.

PCG is differentiated from creativity support tools and mixed-initiative computational creativity by its extremely minimal or non-existent interface with its users and its almost complete disregard for the user's creativity. Most members of the PCG community avoid any direct confrontation with creativity, whether it be human or machine, and instead focus on the practical application and/or breadth of the content that is generated. This focus and limited domain has led to a set of focused content-based evaluation approaches.

Shaker, Togelius, and Nelson describe two general approaches: the top-down and bottom-up [171]. The top-down approach evaluates expressivity by measuring the range and effectiveness of the generation process/product using metrics specific to the process being used (such as linearity and leniency for platforming games). Because of the narrow and detailed domain of the platforming gaming genre, there are dozens of publications on that specific topic. The bottom-up approach directly asks consumers of the generated content for their interpretation, going directly to the product/press source. The inherent interactive nature of games gives users a testbed of use-case scenarios to test. PCG offers a unique example of computational creativity scrutinized by fairly detailed and rigorous metrics.

Compton, Osborn, and Mateas argue for the term "generative methods" instead of procedural content generation to describe any computational approach that results in a contribution to a product or artifact [40]. Like the terms "craft" and "art," it is difficult to expect "content" to cover every conceivable product a computer may contribute to, especially with the implicit focus on game content. Instead, Compton, Osborn, and Mateas shift the focus onto the techniques by which computers generate their products, arguing for a process-centric rather than product-centric approach. The process-centric approach echos the crafting processes framing behind this dissertation.

Liapis, Smith, and Shaker instead apply the "mixed-initiative" label to content creation: explicitly acknowledging that, even when the human just presses a "generate" button, human and machine shared involvement is inescapable in a digital context [112]. The "requirements, caveats, and open problems for mixed initiative systems" outlined by Liapis, Smith, and Shaker that we wish to acknowledge are as follows (page 199-200):

- Who is your target audience?
- What novel and useful editing operations can be incorporated?
- How can the method for control over content be balanced?
- How to resolve conflicts that arise due to the human stating conflicting desires?
- How expressive is the system?
- Can the computer explain itself?

In this context of this dissertation, our audience is domestic amateur textile crafters. The projects cited in this dissertation often include generative methods as a means of supplying content to the user, whether it be as patterns or evaluating game logic. However, many of these projects do not address these open problems, or do so in a very ad hoc manner. An open area of research is to apply evaluation frameworks such as these to experimental games and computational craft projects. It is important to note that this dissertation aligns with Compton, Osborn, and Mateas' view of expanding the definition of what "content" generally refers to. This dissertation considers any product at least partially made by the computer as *content* that was *procedurally generated*, and, following Liapis, Smith, and Shaker, that any generative process that also involves a human is *mixed-initiative*.

Crafting and its historical legacy, as well as examining creativity from a psychological perspective, gives a foundation for modern expressions of craft and creativity throughout this dissertation. In particular, the computer sciences' approach to creativity support helps to generate an understanding of how creativity is ratified within some of the communities related to this dissertation. Of particular importance to the discussion of computer agency is computer creativity and mixed-initiative co-creativity. Communities discussing computer creativity contain mixed opinions on how much credit to attribute to computers versus their human creators and collaborators. It is my perspective that humans not be privileged, and that the contributions of computers to the creative process of humans should not be discredited like so many other disenfranchised groups throughout history.

Chapter 3

Perspective 1: Textile Craft Ideation and Design

One of the long-standing goals of creativity support software research has been to help people express themselves. In instances of high user skill, the user should know what they want to make (expected affordances) and be able to create those designs (via matching presented and perceived affordances) as effortlessly as possible, which is an expression of high digital agency. In instances of low user skill, the user may not know what they want to make (low expected affordances), and that also means they should not presented with many complex tools (fewer presented and perceived affordances) in order to have high agency. Both types of users should also have access to ideation tools that encourage ludic engagement, which, if they exist, are often separate from productivity-based design software that focuses on accomplishing specific tasks. In the case of physical textile crafts, design software involves either a simulation of some aspect of the physical craft, and/or an API that allows people to easily interact with output devices they would otherwise have trouble controlling.

The following chapter is introduced with a continuation of related research into creativity support tools, and a specific sub-genre of playful design software called casual creators. Afterwards, two projects in this genre that implement simulation and API support are presented. Finally, general design insights are summarized from the presented projects for future use by other software designers.

3.1 Design Support Software Research Review

3.1.1 Creativity Support Tools

> Creativity support tools extend users' capability to make discoveries or inventions from early stages of gathering information, hypothesis generation, and initial production, through the later stages of refinement, validation, and dissemination. [175]

These tools — software meant to aid non-digital manufacturing and creativity through idea realization — have been around for decades (Sketchpad in 1963 [193]; WordStar word processor released in 1979 [126]; AutoCAD released in 1982 [17]). A literature review of support tools separated the general-purpose creativity support tasks into four categories: problem finding, information finding, idea finding, and solution finding [211]. While generally phrased as search terms, each of these types of tasks may include creation within their task of searching. The software in this chapter focuses on idea finding, which may bleed into solution finding as techniques for physical manufacturing are explored.

The Conundrum of Productivity-Focused Tools and Evaluation

The at-scale professional and production-focused communities around these types of software highly value practical aspects, such as usability, efficiency, and effectiveness. Usability and efficiency can be somewhat measured empirically using time and tests of understanding or accessibility with regards to different software features. However, *effectiveness* is vague and circumstantial. The difficulty with evaluating the quality of the output has led to the focus on the first two aspects in the form of aforementioned tasks, summarized as *productivity*. Productivity is both extremely valuable to the creativity support software community and a less relevant means of measuring success for the other communities in this dissertation (game design, HCI, and amateur textile crafters). Examining productivity, however, is a useful foil in examining autotelic software below.

Precisely what counts as productivity is directly related to the goal of what the user is making within the software. Some measures of productivity focus on such small-scale problems like using a particular sub-tool (how a pen tool is used to create vectors), while others work on sheer volume of output (how many unique ideas a team was able to generate while brainstorming). In general, productivity is how many times a user can complete their task (or how much progress a user can make toward their task) when using the software and completing a clearly defined task within a given time frame [46]. The measure of productivity has its greatest meaning when compared with other software that is used to complete the same task within the same time frame.

However, even if the software and tasks are the same, the variables within the user are vast. The following is a summary of the challenges, primarily from [176], in getting accurate data on the effectiveness of the computational creativity software:

- The breadth of software specialized toward specific tasks and different domains
- The varied tasks used by different domains within the same software
- The varied familiarity of the user with computers in general

- The user's expertise using that particular (or similar) software
- The user's skill level within their domain regardless of software involvement
- The user's rate or ease of learning between one experiment and the next
- The compound of user-related variation when there are multiple simultaneous users

An Alternative Perspective

Katherine Compton has pioneered the concept of Casual Creators as a "genre of autotelic creativity support systems" in her dissertation and related research [41]. The properties and goals of the genre are expressed in its definition (page 6-7 // 41-42):

> A casual creator is a system that
>
> - privileges enjoyment of the creative process above productivity
> - produces artifacts within a limited-yet-meaningful domain space, enabling automation and support, both passive (encoded into the domain model and system constraints) and active (responding to user actions)
> - supports a state of creative flow by restricting choice and preventing hard failures while allowing rapid iteration
> - results in the user's feeling of pride and ownership toward the produced artifact,
> - and sense of pride in their own creativity.

While most textile craft support tools are not fully classified as casual creators, some of them are [111, 69, 195]).

Examining the ludic engagement and agency with textile design software includes casual creators, especially in the ways that these experiences aim for high ludic engagement. In particular, the creativity software presented in this chapter fulfills the complete definition of a casual creator. While the two projects in this chapter can be used for productivity, the use and outputs of the systems "privilege enjoyment" of their blackwork embroidery and knitting domains. Both software tools represent an "encoded" and "limited-yet-meaningful domain space" of their particular textile craft, while designing around or explicitly eliminating areas of "hard failure" for the creation of digital and physical artifacts that ruins agency. Both systems automate their passive domain restrictions within their domain, while allowing rapid iteration via their simple interface and encoded domain.

3.2 Blackwork Embroidery Pattern Generator

This section (Blackwork Embroidery Pattern Generator) is adapted from the previously published paper Blackwork Embroidery Pattern Generation Using a Parametric Shape Grammar *with accompanying authors Noah Wardrip-Fruin and Michael Mateas (view the full citation here: [78]). The work presented here is entirely my own.*

This project presents an interactive parametric shape grammar for blackwork embroidery pattern generation, whose patterns are then implemented (sewn) using an unmodified home embroidery machine (see Perspective 2 for the sewn output). A design tool executes the grammar-guided user input and enumerates expanded pattern possibilities at the push of a button. The grammar is capable of generating previous embroidery patterns published by professionals, as well as a huge number of possible new patterns. This design tool prioritized the creativity support tool goal of rapid exploration of the design space (section 2.2.1). Due to the ease of use, freedom of exploration, and surprise of the output in the design tool, it enables ludic engagement. Idea iteration, while not necessarily increasing agency with the tool itself, enables increased feelings of agency when it comes to crafting the designs shown in the tool.

The main contributions of this work are:

(BW 1) representing counted-stitch embroidery digitally in a more feature-rich format than other grid-based embroidery design software;

(BW 2) exploring local properties of stitches to develop into grammar rules capable of generating previously published blackwork embroidery patterns as one possible measure of quality;

(BW 3) offering those stylistically different grammar rules as the perceived affordances of our system to a designer of embroidery patterns; that provided as instant visualization of enumerated pattern possibilities;

In order to accomplish these objectives in the following sections, I will introduce related embroidery and shape grammar history, give an overview of our architecture, discuss the expressive space of our current system, and demonstrate the validity of its generated patterns. A sample of our system's generated designs can be seen in Figure 3.1.

3.2.1 Blackwork Embroidery Related Work

In the multi-millennial lifespan of embroidery, while many fundamental stitches remain the same, dozens of styles and approaches have been classified [109]. This project chooses to focus on non-freeform blackwork embroidery, one of the most restrained styles, as a first approach to embroidery pattern generation.

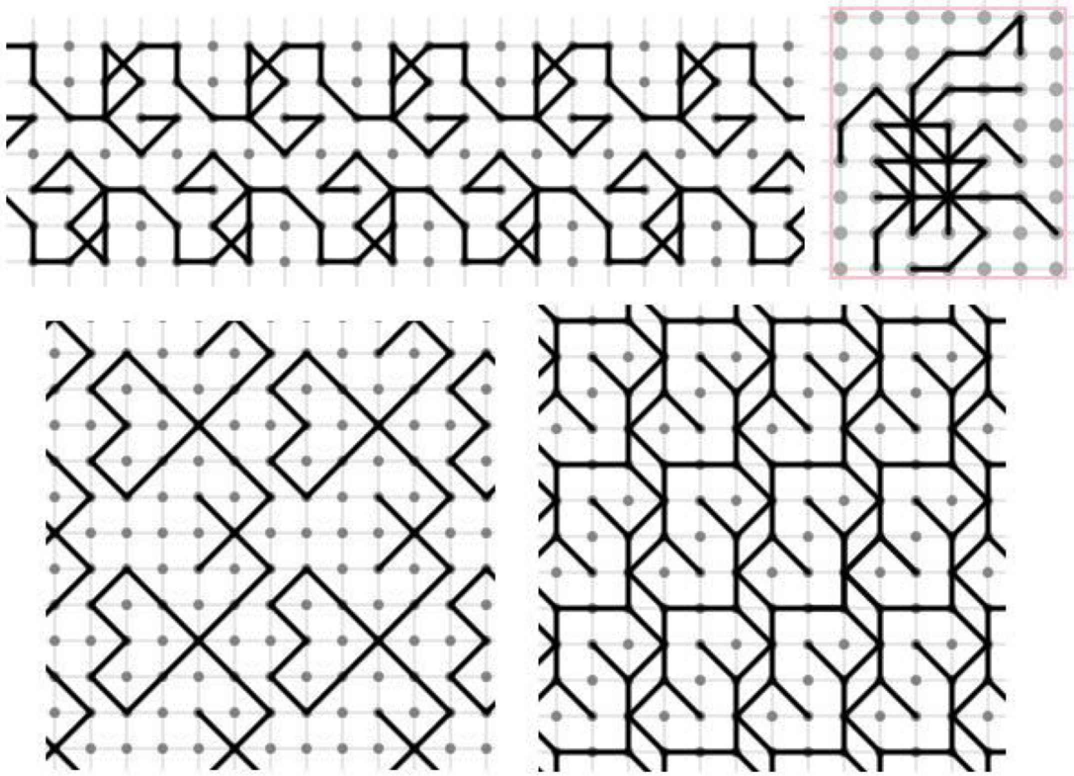


Figure 3.1: Generated samples by our parametric shape grammar. These demonstrate a border (upper left), a focal piece (motif, upper right)), and two examples of space-filling patterns (lower left and lower right). The motif has a highlight to distinguish it as the seed for further post-production techniques.

Blackwork as counted-stitch black embroidery on white fabric has a mixed cultural heritage, but it was popularized in 16th century Tudor England, as seen on the caps, cuffs, and coifs of people in paintings from that time [72] (see Figure 3.2). The current classification of blackwork, in spite of its name, is not always worked in black nor in a single color. Blackwork, sewn using back stitch and double running stitch, is characterized by dense, complex, and self-similar geometric shapes.

Joshua Holden has developed mathematical graph proofs on solving reversible hand sewing approaches to blackwork designs based on these stitches [84]. However, our system focuses on visualizing and designing new patterns for a human or embroidery machine to interpret. Existing patterns designed by humans gen-

Figure 3.2: 16th century Tudor blackwork [216]

erally come in three forms: to be used alone (as a motif or focal point), repeated horizontally or vertically (as edges or borders), or horizontally and vertically (as space-fillers). These three forms are demonstrated in Figure 3.1.

It is often tedious and troublesome to create fresh patterns, where an author would carefully transcribe dozens or hundreds of stitches rotated or duplicated across a motif or pattern. Procedural pattern generation alleviates these boring and repetitive pattern design tasks, offering the designer instant visualization of alternative patterns or applications, such as turning a motif into a fill through many repetitions. Digital representations also allow users to design patterns without committing the time or effort to draw or sew them out, which is arguably the biggest strength of design ideation tools (discussed more in sections 2.2.1 and 3.1.1).

Shape Grammars

Originally formalized by George Stiny, shape grammars are systems of transformation rules that change one shape into another [186]. Shape grammars belong to a class of techniques involved in procedural generation and have been applied to many artistic and technical topics, from office chairs [85] and motorcycles [149] to art [187, 101]; and architecture [104, 31, 218]. Parametric shape grammars extend regular shape grammars to be more sensitive to their surroundings, and allow us to more intelligently select which shape rules to expand. As a first draft for this approach, our parametric expansion rules are based off of local density and visual connectedness of stitches (within two stitches) and are detailed below.

3.2.2 Architecture

We have chosen an 8-way linked list as a graph structure to represent the sewing space, with active graph edges representing the stitches and the nodes representing holes in the fabric at the end points of stitches. The final pattern can be sewn using a graph traversal algorithm, which we know will obey the spatial properties of our physical fabric. The graph pattern is laid down over a grid of arbitrary size, representing any scale of evenweave[1] fabric the crafter may use. This representation accomplishes our (BW 1) goal, as automatically scaling machine or hand embroidery designs is not a trivial task for broad spectrum embroidery design software. Figure 3.3 presents a high-level view of the architecture and user workflow while using our tool. Each of the numbered stages represents a phase of

[1] Fabric where the warp and weft (horizontal and vertical) threads are evenly spaced and the same size, which forms a perfect grid.

building up the complexity of the design from starting line (Figure 3.3 Stage 1) to output readable by an embroidery machine (Figure 3.3 Stage 5, which will be covered in more detail in section 4.2.1). In stages 2-4, the generation is driven by the author and generated/visualized by the tool for rapid development.

Stage 1: Start

To help initiate creativity and avoid the blank canvas problem, the architecture always seeds the generative space with a line of a random orientation for the user to generate from (Figure 3.3 Stage 1) [205]. To help support the author, the following rules and transformations are applied via clicks of buttons, so prior stitch design expertise is not required to make elaborate patterns.

Stage 2: Grammar Expansion Rules

Aligning with our octal linked list data structure, there are eight possible directions of expansion from a single point: up, down, left, right, upper-right, upper-left, lower-left, and lower-right. A single application of any grammar rule transforms a node into a node-and-line extending out in one of these eight directions, generating an edge in the graph and another point if necessary. At each request for a rule application, the system generates every possible stitch growth on the current design and ranks them based on the expansion protocols below. Depending on the protocol, the system picks either the top rule, a weighted random selection, or a completely random selection and executes the rule. The “Forbid Crossing Diagonals” button removes any expansion possibilities that would cross an existing design line. Forbidding crossing diagonals is a design choice that separates blackwork from the cross-stitch domain of embroidery. It also eliminates (or allows) additional density.

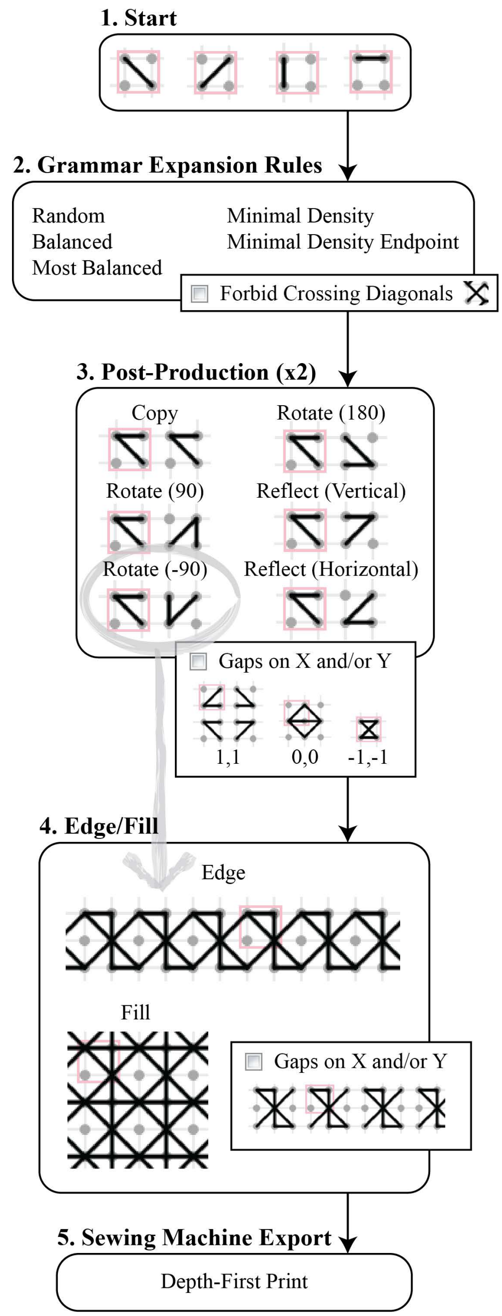


Figure 3.3: An architecture overview and workflow pipeline, with stages labeled 1 through 5. The rectangular sub-boxes represent additional constraints or options that the author may specify. The highlighted Post Production method in stage 3 is the one selected for expansion demonstration in stage 4. Stage 5 will be covered in more detail in section 4.2.1.

All of the generated samples in this blackwork project have not been hand-altered in any way in order to preserve the authenticity of the claims in this section.

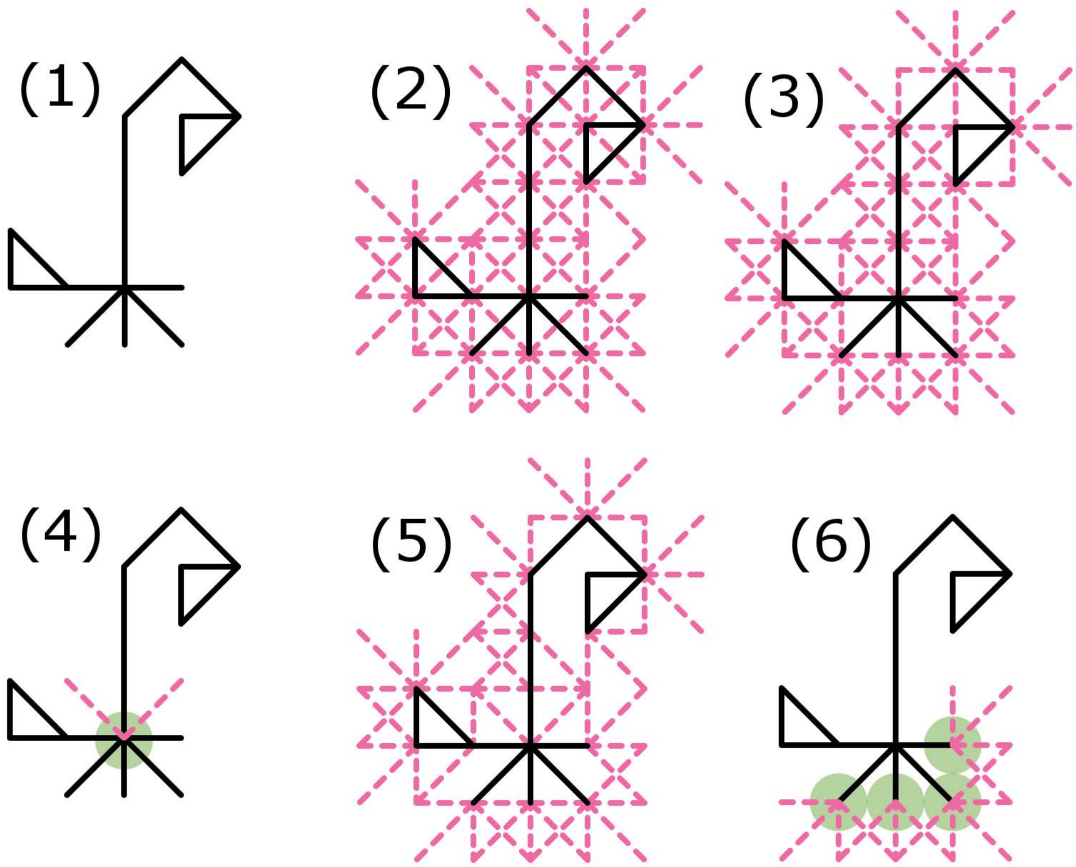


Figure 3.4: Suppose (1) is the design in its current state. (2-6) demonstrate all the possible expansion possibilities on the next expansion step depending on the expansion rules, which are shown in pink dotted lines. When an expansion rule ranks possibilities, their highest ranked options are highlighted with green circles along with the pink dotted lines. (2) shows all valid expansion rules that can be chosen by Random Expansion; (3) shows (2)'s lines minus Crossing Diagonals on the existing design in (1). (4) highlights the most dense point on the design, so expansion strategies that maximize density will rank the two lines connected to that point highly. (5) shows (3) without dense endpoints âĂŞ that is, without lines that would create another cycle in the graph. (6) show the most highly ranked expansion possibilities based on how minimally dense the existing design is. The highlighted points are the minimally dense points (with only 1 existing connection), so the pink dotted lines are the most highly ranked options.

Random Expansion

The rankings of rules are fundamentally ignored in Random Expansion, and any valid rule is chosen in this expansion protocol (Figure 3.4 (2)). Validity requires only that the line does not currently exist. However, if the author has the "Forbid Crossing Diagonals" checkbox checked, a grammar rule is also invalid if it creates a diagonal line that would cross another existing diagonal line (Figure 3.3 (2) Forbid Crossing Diagonals causes Figure 3.4 (3)).

Density-Based Expansion

Density is a score applied to each node that represents how many lines are connected to it. Every new line increases the density of two nodes. The Minimal Density expansion protocol selects lines that increase the design's density the least (Figure 3.4 (6)). A design following only minimal density appears as a meandering line. To take additional advantage of our density scores, we offer a Minimal Density Endpoint expansion protocol, which only takes into account the density of the far end of the new line (Figure 3.4 (5)).

Balanced-Based Expansion

Balance is another approach to analyzing the properties of a node and its edges. A node's balance is dictated by how many instances of symmetry it currently has. For example, a node with all possible eight lines attached has maximum balance. Any line that increases the balance of any node is considered as part of the Balanced expansion protocol, while a line that increases the balance of both line endpoints is preferred by the Most Balanced protocol. Balanced formations generate tight stars and would, for example, favor the new lines in Figure 3.4 (4) or the lines that were excluded in Figure 3.4 (5).

Stage 3: Post-Production Transformations

Because nearly every blackwork pattern researched for this project contained some form of self-similarity, and that would be extremely difficult to ensure with the grammar alone, we separated the design of pattern section repetition into a separate phase. Post-production has two steps: duplication, reflection, or rotation on the horizontal and vertical axes. The user also has the option of offsets for these blocks: to allow them to overlap or be given space. The post production transformations are shown in Figure 3.3 (3).

Stage 4: Edge/Fill Visualization Techniques

The eventual application of most modern blackwork patterns are as edges (around a picture, cuff, or collar) or as fills (to repeat and fill out an enclosed space). Both example uses can be seen in Figure 3.2. Because of the modular creation of our design and its post-production, it was an easy expansion to repeat the design horizontally and vertically to visualize these eventual uses for the patterns. Optional gaps or overlaps can be added between the whole of the design, similar to how post-production added gaps between the transformations.

3.2.3 Blackwork Grammar Discussion and Evaluation

One measure of our success of goal (BW 2), the capability to generate previously published blackwork embroidery patterns solely with our expansion rules, was either trivially simple or exceptionally challenging, depending on the design. For fill patterns especially, some designs are repetitions of tiny elements, so it was easy to match a few previously published hand-authored instances of patterns. However, the possibility space for stitch configuration is exceedingly rich, so it was very challenging to exactly match published patterns that were made up of

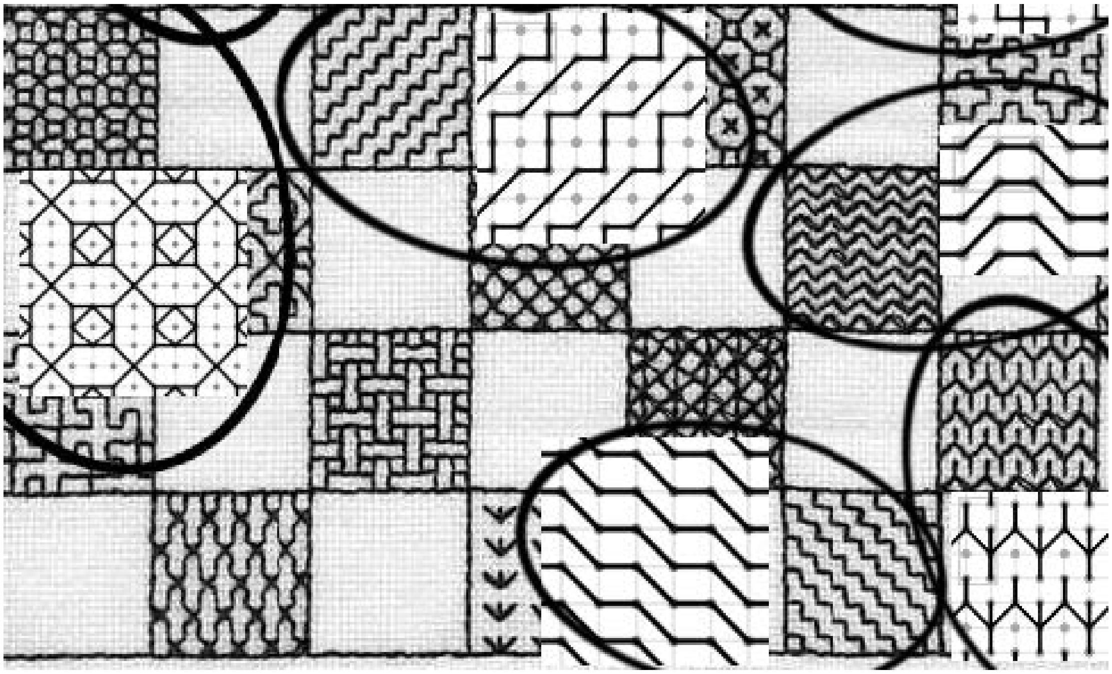


Figure 3.5: Part of a modern blackwork embroidery sampler, which we have altered to highlight pairs of the original fill patterns (the smaller of the pair) and samples made with our system (the larger of the pair) [216].

more than a handful of stitches. However, we did generate very similar designs to many we have seen (Figure 3.5). Further samples of matched patterns can be found in publications such as [81] and [83]. The patterns generated by the grammar exceeded our expectations, as well as those sewers and naive users to whom we informally demoed the system.

A formal user study was outside the scope of this project, but the various forms of grammar rule expansion and postproduction did offer distinctly different styles of patterns. Minimal density offers open and loose patterns, while most balanced expansion creates great density. As for demonstrating our visualization usability goals in (BW 3), feel free to demo our system online or see it in the supplementary materials. Exploring each of the expansion rule choices in depth should offer a feel for how they affect the design, which should gradually increase the comfort and presented and perceived affordances while interacting with the tool. While the user interface is simple, with familiar web design elements of buttons and number spinners, the academic descriptions of what each button does was not helpful and resulted in low perceived affordances. The primary take-aways of the user interface feedback I received to increase presented affordances was to come up with simpler names, or forgo names entirely and instead offer alternative parallel visualizations of what each button would result in if it was pressed.

Because our embroidery surface was represented as a graph, we decided to investigate some common graph algorithms to see if they could aid with pattern and generative evaluations. While we did examine minimum spanning trees and cliques, they ended up not being as practical or understandable as those embroidery properties we developed and outlined above (density and balance).

3.2.4 Blackwork Embroidery Conclusions

This section demonstrates our research contributions by developing a web-based design tool capable of outputting designs for hand or machine embroidery. The parametric shape grammar generated previously hand-authored patterns, as well as new designs using our grammar expansion rules. These grammar rules were applied to produce visually distinct patterns. Similar crafts that use motif repetition and tessellation, such as wallpaper, fabric, or surface pattern design, can also make use of this grammar, design principles, and the pipeline presented in this project. While there are more usability improvements and feature expansions to add, we have a solid and efficient architecture using novel applications of digital graph algorithms for physical crafts.

This design tool is a creativity support tool, one that rapidly explores the design space of blackwork embroidery and empowers the user with machine and hand embroidery designs at the end of use. The familiar user interface, quick and drastic visualization of alternative designs, and abstraction of tedious blackwork style design elements smooth the experience and support unobstructed ludic engagement. The design tool's perceived affordances are few due to it being a barebones support software (low presented affordances), which leads to unbalanced and low digital agency for users with high expected affordances that are looking for the features of a full embroidery editor. However, unexperienced users approached with low expected affordances and had more digital agency when interacting with the tool. Finally, the tool's direct relation to real-world patterns that the user can sew, as well as automatic stitching via an embroidery machine, increases feelings of physical agency with the domain as well.

The next design tool in this chapter tackles a much more difficult domain: 3D machine knitting. The knitting compiler project prioritizes the abstraction of confusing and potentially dangerous design decisions when working with expensive machines, granting users confidence and safety to explore ludic engagement. At the same time, the design tool abstractions match user's expected affordances of knitting operations (rather than machine knitting operations) much better, which makes the presented and perceived affordances of software-based knitting design much easier to achieve. Overall, the likelyhood of higher digital agency is much higher.

3.3 3D Machine Knitting Compiler

This section (3D Machine Knitting Compiler) is adapted from A Compiler for 3D Machine Knitting, *a collaborative article with accompanying authors James McCann, Lea Albaugh, Vidya Narayanan, Wojciech Matusik, and Jennifer Mankoff, and Jessica Hodgins (for full citation, see [124]). My contributions to this work primarily focused on the user experience via the 3D interpretation and visualiza-*

tion of the generated knit patterns.

Figure 3.6: Our compiler processes high-level primitives into low-level instructions for production on industrial knitting machines.

While industrial knitting machines are not typically amateur, they[2] are accessible to the home market for smaller-scale production similar to embroidery machines. These industrial knitting machines can produce finely detailed, seamless, 3D surfaces quickly and without human intervention. However, the tools used to program them require detailed manipulation and understanding of low-level knitting operations. We present a compiler that can automatically turn assemblies of high-level shape primitives (tubes, sheets) into low-level machine instructions. These high-level shape primitives allow knit objects to be scheduled, scaled, and otherwise shaped in ways that require thousands of edits to low-level instructions.

At the core of our compiler is a heuristic transfer planning algorithm for knit cycles, which we prove is both sound and complete. This algorithm enables the translation of high-level shaping and scheduling operations into needle-level operations. The following work on this project also shows a wide range of examples

[2]Both the knitting machine used in this paper, as well as simpler models.

produced with our compiler and demonstrates a basic visual design interface that uses our compiler as a backend.

Abstracting away these low-level instructions not only reduces the intellectual burden on the designer, but the algorithm verifies their correctness, which leaves the designer confident that they can indeed ignore the low-level details. Reducing the burden on the designer means that they require less domain-specific knowledge, which not only simplifies the design tool, but encourages less knowledgeable amateurs to approach machine knitting as a crafting domain. The lower barrier to entry and safety in design exploration enables increased ludic engagement. In addition, the abstracted layer of knitting operations more closely matches real-life knitting equivalents; a hand-knitter does not know what a tuck is, but they do know what increases and decreases are, which involve tuck operations on a knitting machine. This allows digital agency to better align with physical agency, and should increase the designer's digital agency by offering more appropriate and familiar presented and perceived affordances to match their expected affordances.

Machine knitting is a mature fabrication technology, used to create items ranging from gardening gloves to fashionable sweaters. Knitting machines are programmable, general-purpose devices; however, they are used almost exclusively to manufacture a fixed palette of pre-programmed objects, occasionally with some customization of color patterns. No knit shop today approaches the flexibility common to CNC[3]-on-demand machine shop operations.

This lack of flexibility is a consequence of current knit design tools. The industry standard tools for machine knitting [170, 188] provide high-level templates for a few standard objects, but otherwise leave the user to manipulate needle-level control instructions in a way that fails to divorce machine-specific details from actual fabrication operations. This situation is similar to requiring all CNC machine

[3]Computer numerical control.

users to write toolpath G-code by hand, or all computer programmers to work in assembly.

The main contributions of our work are:

(KC 1) A knitting design representation consisting of generalized tubes and sheets, with gluing instructions at their boundaries, which allows high-level schedule and structure manipulation.

(KC 2) A knitting assembly language that formalizes the low-level operations used by industrial knitting machines to construct knitted objects.

(KC 3) A compiler that can transform the former representation into the latter, at whose core is a complete transfer-planning heuristic for cycles (with associated correctness proof).

3.3.1 Knitting Compiler Related Work

Our work sets out to offer knit designers and programmers a better choice of primitives to use for controlling their output device. We take inspiration from work in rendering, where the primitives have been tailored to output modality (e.g., Reyes [43] for offline rendering and OpenGL for real-time rendering), and from ongoing work in 3D printing, where the community is actively developing and refining primitives (e.g., [208]). Our knitting primitives have orthogonal scheduling and shaping degrees of freedom, inspired by the Halide system [151], in which algorithms are treated as having separate definitions and schedules.

Most prior work surrounding textile design assumes a "cut-and-sew" approach, where garments are made from flat sheets of fabric, cut by humans or machines, and sewn by humans. This area is well-covered [114], with commercial systems widely available [88], and active research supporting sketched input [87], situated interaction [215], and advanced simulation during interaction [201].

One of the great advantages of knit fabric, however, is that it need not be locally flat. This flexibility presents new specification challenges, which are not well-addressed by current tools. Knitting design systems from machine manufacturers [188, 170] provide detailed machine-level control languages (SINTRAL and KnitPaint, respectively) and some macro features that can be used to ease repetitive tasks (e.g., in hand-creating a library of shaped parts [202]), but little in the way of general high-level primitives. Third-party commercial design tools are limited to texture and color design on flat panels [116]. In the research sphere, Knitty [87] provides sketch-based design with tube primitives for hand knitting; it would be interesting, though non-trivial, given the limitations of knitting machines, to retarget that system's output to our machine-knitting backend.

Recent advances have made knit simulation both tractable and predictive [95, 96, 34, 35]. However, setting up initial yarn paths can be tedious. One option is to use visually reasonable (but not feasibly knittable) paths [219]. Our knitting assembly language provides another option: one could create an interpreter to run the language and output virtual yarn paths for a simulation system. Such an interpreter would be able to create simulation descriptions for literally anything a knitting machine could make, using the same instructions as the machine.

3.3.2 An Abstract Knitting Machine

Our compiler targets a knitting assembly language which captures the capabilities common to industrial knitting machines, while abstracting mechanical details that may change between them. We define this language in terms of the actions of an abstract knitting machine.

Knitting machines build their output by manipulating loops of yarn. Consider these loops of yarn (Figure 3.7):

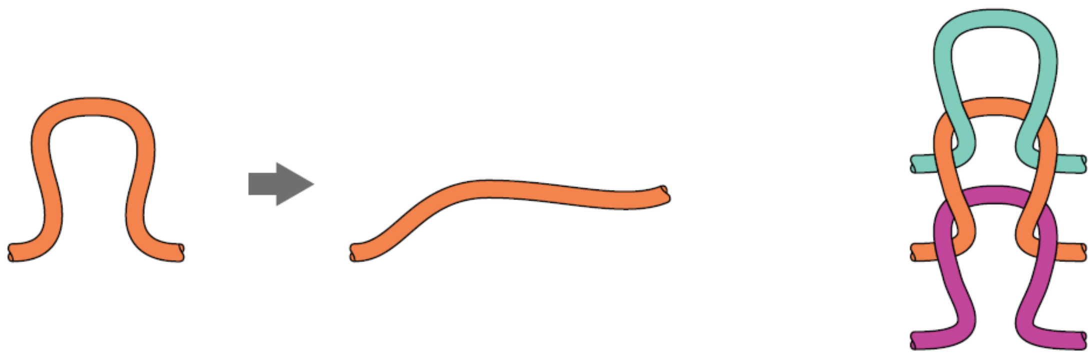


Figure 3.7: The orange loop on the left unravels, while the orange (middle) loop on the right is stable.

The orange loop on the left is not stable: pulling on either end of it would unravel it into a straight piece of yarn. However, the orange loop on the right is stable because it passes through and around other loops.

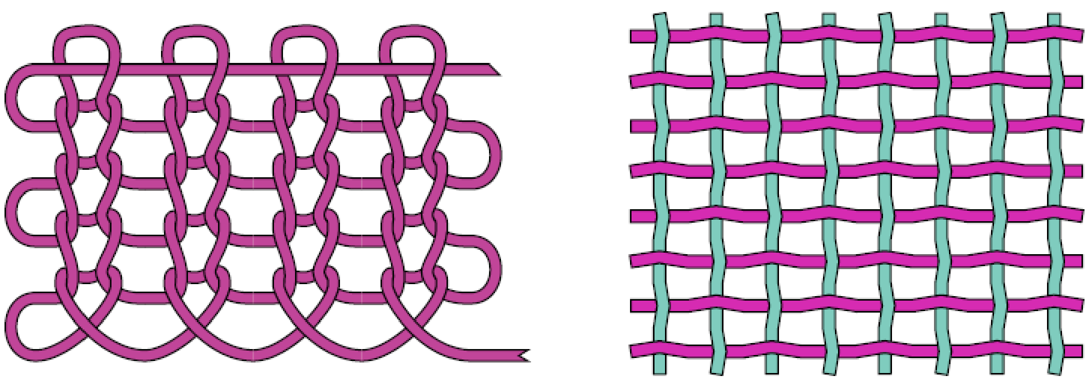


Figure 3.8: The left fabric is knit, while the right fabric is woven (as it might be on a loom).

This "loops through loops" architecture is the basis of knit items. Notice how different the structure is from typical woven items, which use a "yarn over/under yarn" architecture (Figure 3.8).

Machine

Knitting machines hold loops on hooks called needles. These needles are arranged into rows called beds. Figure 3.9 shows a sample bed of five needles.

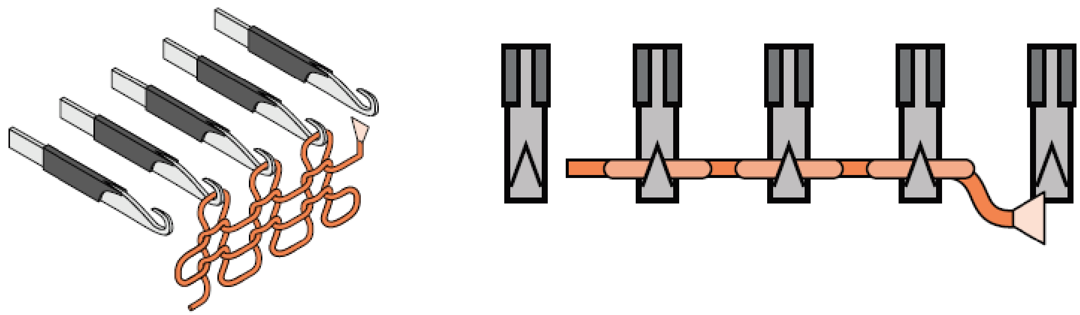


Figure 3.9: The left is an isometric view of a bed of five needles, and the right is a top view of the same bed.

V-bed knitting machines have two beds whose needles face each other. These are referred to as the back bed and front bed. Having two beds allows the machine to hold tubes as well as sheets (Figure 3.10).

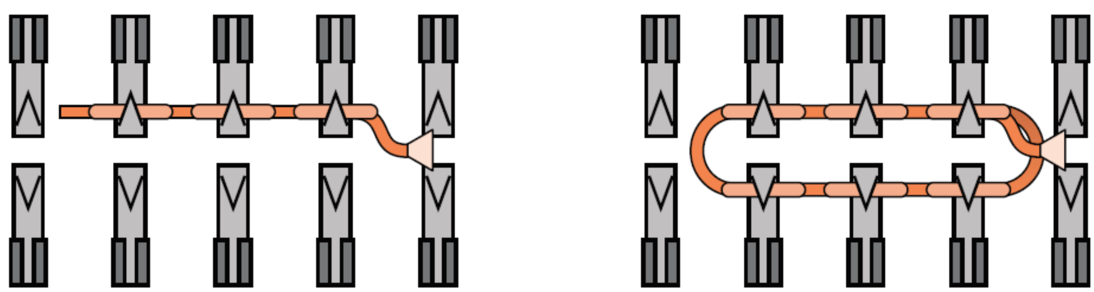


Figure 3.10: A top view of 5-needle back and 5-need front beds.The left view shows a sheet of fabric, while the right view shows a tube of fabric.

Yarn enters the machine from a cone, passing through a tensioning device and a yarn carrier on its way into the knit object. Yarn carriers move laterally between the beds, positioning new yarn where it is needed. There is one yarn carrier for every yarn in use on the machine. In our diagrams, we draw yarn carriers as small triangles between the beds.

Machines create knit objects by manipulating the loops held on their needles. Needles can perform four basic operations: tuck adds a loop to those the needle is holding; knit pulls a loop through all the loops the needle is holding while releasing them; transfer hands all the loops a needle holds to another needle; and split is a combination of knit and transfer that passes a new loop through all the loops a needle is holding while moving them to another needle.

Knitting Assembly Language

We formalize the above operations as a knitting assembly language, which our compiler targets. A backend then further translates these instructions into a machine-specific format.

We begin by defining identifiers for each needle:

$$\forall i \in \mathbb{Z} : \begin{cases} b_i : \text{back bed needle} \\ f_i : \text{front bed needle} \end{cases} \tag{3.1}$$

Needle indices run left-to-right along a bed, and are aligned front-to-back. So f_{-2} is aligned with b_{-2}, which is three needles to the left of b_1. Although needles often hold only one loop, they can hold several at once; thus, when talking about needle operations, we will write $n_i = [l_1, ..., l_t]$ to indicate that loops $l_1, ..., l_t$ are held by needle n_i,with it being closest to the tip of the needle. We will also occasionally (for notational convenience) conflate needle locations with their integer indices, writing such phrases as $f_2 - b_0 = 2$.

We endow our abstract machine with a set of active yarns Y, which starts empty, and limit it with a maximum racking (lateral bed offset) value of R.

We abstract the motion of the yarn carrier by introducing a primitive to create loops: Let $\texttt{loop}(y, d, n)$ where $y \in Y, d \in \{+, -\}, n \in \{f_i, b_i\}$ return a new loop created by passing yarn y in direction d over needle n.

Here we illustrate and define each of the four operations for a standard knitting machine:

Tuck. The *tuck* operation adds a new loop of yarn l_{n+1} in front of the loops $[l_0, ..., l_n]$ already held on a needle. Mechanically, the needle reaches forward, the yarn carrier moves to the right over the needle, and the needle retracts, now holding a new loop. We illustrate this in isometric and top views in Figure 3.11:

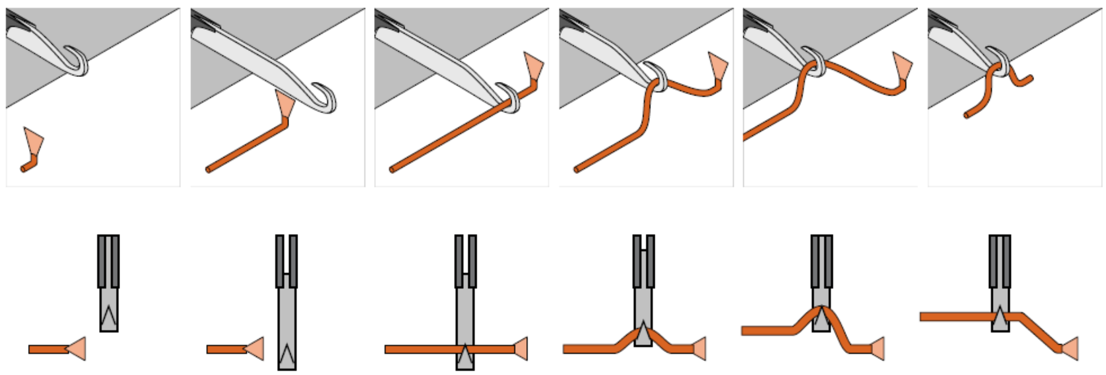


Figure 3.11: The top row shows the isometric view of the needle bed during the tuck operation. The bottom row shows the top view of the needle bed during the tuck operation.

Tucking a needle already holding a loop stacks a new loop in front of the old loop (Figure 3.9):

The yarn carrier moved to the right over the needle in the example above, so this was a "tuck right." Tucks can be formed to the left or right, regardless of where the yarn was previously used. Here, the yarn was last used to the right of the needle, but the carrier can still be moved to the left, and then the needle tucked right:

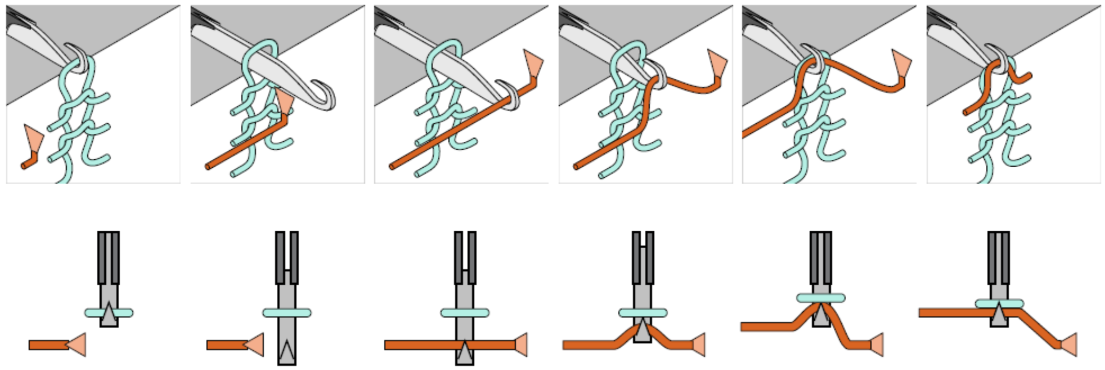


Figure 3.12: The top row shows the isometric view, and the bottom row shows the top view of the needle bed during the tuck operation when there is fabric currently on the needles.

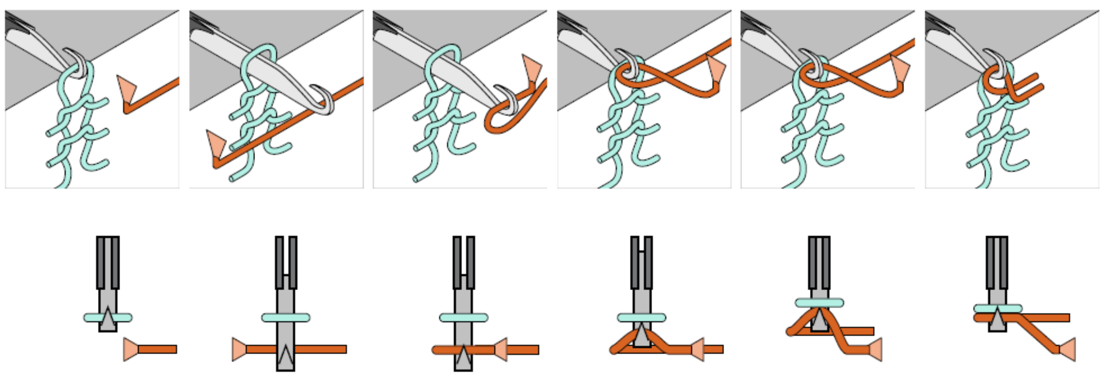


Figure 3.13: The top row shows the isometric view, and the bottom row shows the top view of the needle bed during a right tuck when the yarn carrier draws yarn from left of the needle.

Mathematically, we define tuck as follows:

$$\underline{\texttt{tuck}\ y, d, n}$$

$$\textbf{Given: } y \in Y, d \in \{+, -\}, n \in \{f_i\} \cup \{b_i\} \tag{3.2}$$

$$n \leftarrow \text{cat}(n, [\text{loop}(y, d, n)])$$

Where "cat" is a function that concatenates lists.

Knit. *Knit*ting a needle pulls a new loop of yarn through *all of the loops* currently held by that needle. Mechanically, the needle reaches forward, the yarn carrier moves over it, and the needle retracts, using a secondary mechanical action to lift the loops that it was holding up and over the new loop and off of its tip (Figure 3.14).

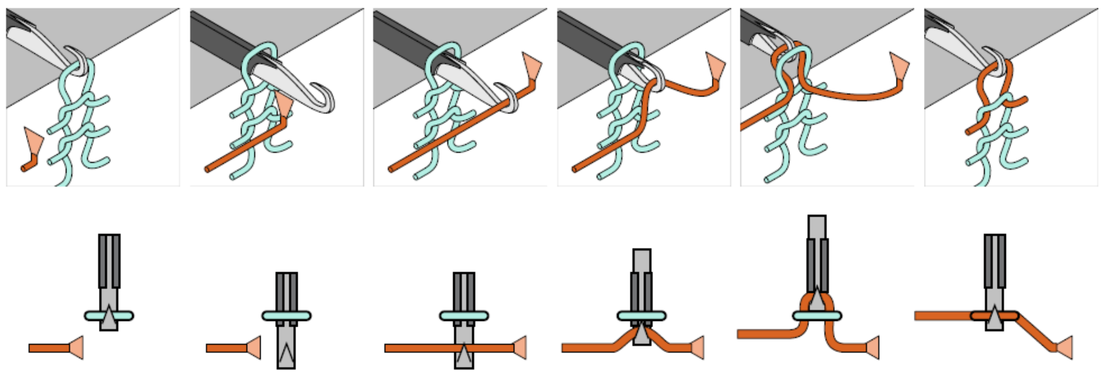


Figure 3.14: The top row shows the isometric view, and the bottom row shows the top view of the needle bed during a knit operation. The new (darker, orange) yarn carried by the yarn carrier is the only yarn left on the needle after the operation.

Knit, like tuck, has a direction. The above example is a "knit right" because the yarn carrier moves to the right when supplying the yarn for the new loop.

$$
\begin{aligned}
&\underline{\texttt{knit}\ \ y, d, n} \\
&\quad \textbf{Given: } y \in Y, d \in \{+, -\}, n \in \{f_i\} \cup \{b_i\}, n \neq [] \\
&\quad l \leftarrow \text{loop}(y, d, n) \\
&\quad \text{pull}(l, \text{reverse}(n)) \\
&\quad n \leftarrow [l]
\end{aligned}
\tag{3.3}
$$

Where "pull" means to pull a loop through a list of other loops; and "reverse" reverses the order of a list.

Transfer. The *transfer* operation moves all the loops on a needle to the needle across from it. That is, it moves loops from the front bed to the back bed or visa versa (Figure 3.15).

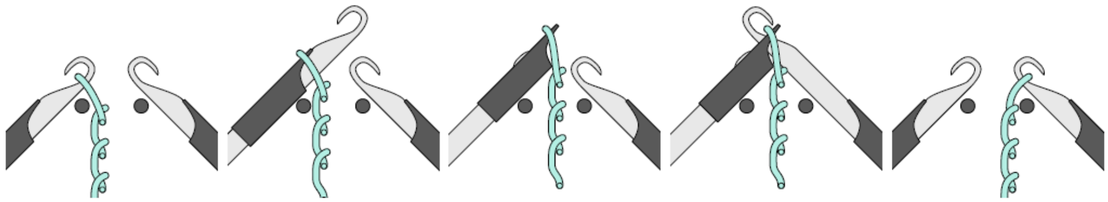


Figure 3.15: The side view of one transfer operation, where the yarn hanging from one needle is passed to another needle across from it on the other bed.

$$\begin{aligned}
&\underline{\texttt{xfer}\ n, n'} \\
&\quad \textbf{Given: } (n, n') \in \{(f_i, b_j), (b_i, f_j)\}, |i - j| \leq R \\
&\quad n' \leftarrow \text{cat}(n', \text{reverse}(n)) \\
&\quad n \leftarrow []
\end{aligned} \tag{3.4}$$

This restriction of only moving between aligned needles may seem severe, but machines can *rack* (laterally move) the beds to change which needles are aligned. By convention, we take racking values as the offset of the back bed (Figure 3.16):

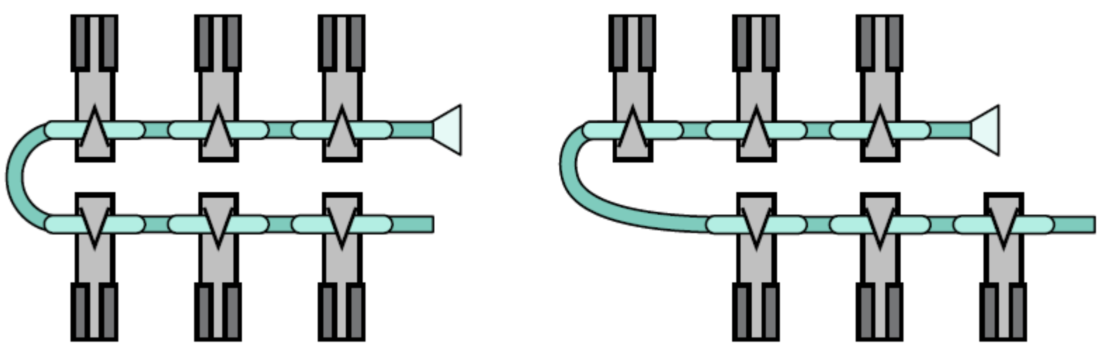


Figure 3.16: The top view of two needle beds. The top needle bed is the back, while the bottom needle bed is the front. The image on the right shows back bed is shifted (rack = -1) compared to the image on the left (rack = 0).

Split. The *split* operation combines knit and transfer into one operation. Split is useful because it allows the machine to knit through a loop without losing the ability to access the loop in the future (Figure 3.17):

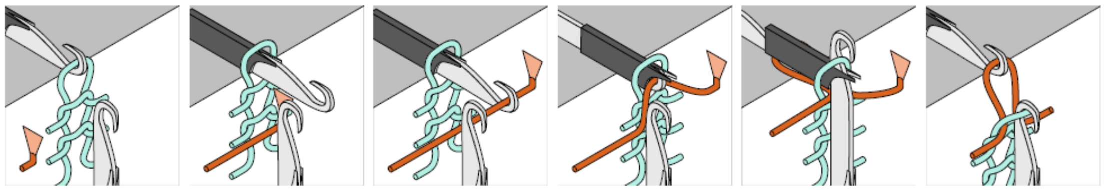


Figure 3.17: Illustrates a right split operation. The needle on the opposite bed hooks onto the lighter, pale blue knit stitch before the original needle pulls the carrier yarn (orange, darker) through the blue thread as a knit operation.

Like knit, split has a direction. The above is a split right.

$$\begin{aligned}
&\underline{\texttt{split}\ \ y, d, n, n'}\\
&\quad \textbf{Given: } y \in Y, d \in \{+,-\}, n \neq [],\\
&\quad (n, n') \in \{(f_i, b_j), (b_i, f_j)\}, |i-j| \leq R,\\
&\quad l \leftarrow \text{loop}(y, d, n)\\
&\quad \text{pull}(l, \text{reverse}(n))\\
&\quad n \leftarrow \text{cat}(n', \text{reverse}(n))\\
&\quad n \leftarrow [l]
\end{aligned} \tag{3.5}$$

Finally, we introduce three utility instructions that are important for yarn management and finishing:

Drop. The instruction *drop* causes a needle to drop the loops it is carrying. Mechanically, this is *knit* with no yarn from the carrier:

$$\begin{aligned}
&\underline{\texttt{drop}\ \ n}\\
&\quad \textbf{Given: } n \in f_i, b_i, f_i^h, b_i^h, n \neq []\\
&\quad n \leftarrow []
\end{aligned} \tag{3.6}$$

In, Out. The instructions *in* and *out* add and remove active yarns. When a yarn is removed, the connection between it and its last stitch is broken.

$$
\begin{aligned}
&\underline{\texttt{in}\ y}\\
&\quad \textbf{Given: } y \notin Y, y \text{is a yarn}\\
&\quad Y \leftarrow Y \cup \{y\}\\
&\underline{\texttt{out}\ y}\\
&\quad \textbf{Given: } y \in Y\\
&\quad Y \leftarrow Y \setminus \{y\}
\end{aligned}
\tag{3.7}
$$

A Slight Extension

For convenience, we describe our algorithms in terms of a more advanced form of v-bed knitting machine called an x-bed machine. This machine adds an extra mechanical element (called a *holding hook*) above every needle in the front and back beds. These holding hooks can hold loops, but cannot knit or tuck. If any loops are held on the holding hook associated with a needle, that needle cannot be used to perform an operation, as the held loops will block it (Figure 3.18):

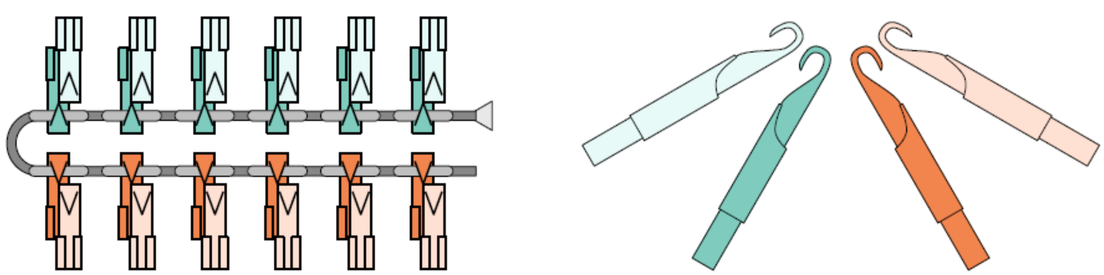


Figure 3.18: To the right is a top view of an x-bed machine. All of the holding hooks are holding loops which physically interfere with the needles operating. To the left is a side view of how the holding hooks (shaded darker) are below their accompanying needles.

For an x-bed machine, we need to add two new types of hooks to the array of needles:

$$\forall i \in \mathbb{Z} : \begin{cases} b_i : \text{back needle} \\ b_i^h : \text{back holding needle} \\ f_i^h : \text{front holding needle} \\ f_i : \text{front needle} \end{cases} \tag{3.8}$$

Finally, because the use of the holding hook associated with a needle limits the use of that needle, we need to make it a condition of any knit, tuck, transfer, or split operation that all of the needles involved are *clear*. We say a needle n is clear if it is a holding hook or if its associated holding hook is empty:

$$\begin{aligned} & && n = f_i^h \vee n = b_i^h \\ \text{clear}(n) \equiv \exists i : & \quad \vee && (n = f_i \wedge f_i^h = []) \\ & \quad \vee && (n = b_i \wedge b_i^h = []) \end{aligned} \tag{3.9}$$

What Knitting Machines Can Make

We now show how the operations described in the previous section can be used to make and deform 3D shapes. These techniques are the core of the generalized shape primitives supported by our compiler.

Sheets can be created by knitting back and forth over a needle range; e.g., knitting right on $b_{-5}, ..., b_5$, knitting left on $b_5, ..., b_{-5}$, and repeating five times will create a sheet 11 stitches wide and 10 courses (rows of knitting) tall. Tubes can be created by knitting in a consistent direction around a circle of needles; e.g., knitting right on $b_1, ..., b_{10}$, knitting left on $f_{10}, ..., f_1$, and repeating 12 times will create a tube 20 stitches in circumference and 12 courses tall. These shapes can be further modified with *shaping* operations.

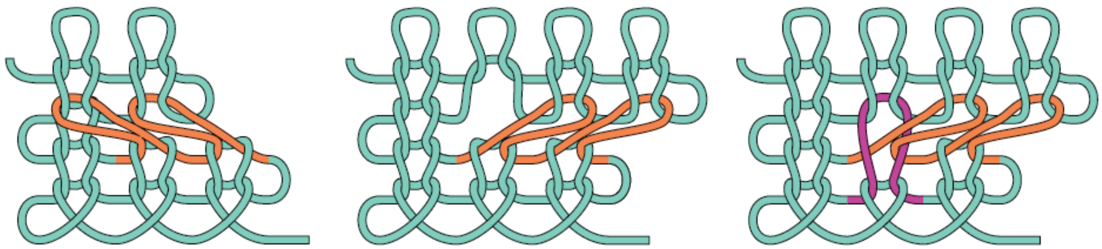


Figure 3.19: Increase and decrease shaping can change primitive width, where the tuck increase (middle) leaves a noticeable gap.

Increases and decreases (Figure 3.19) can be used to adjust the number of stitches in a course, allowing tubes (sheets) of continuously varying circumference (width). The width of a course of stitches can be *decreased* by moving a stitch onto the same needle as its neighbor, and knitting both of them with a single stitch in the next row. Conversely, the width of a course can be *increased* by moving stitches apart and filling this gap with a tuck in the next course. A visually more pleasing increase can be performed by using a split instead of a tuck.

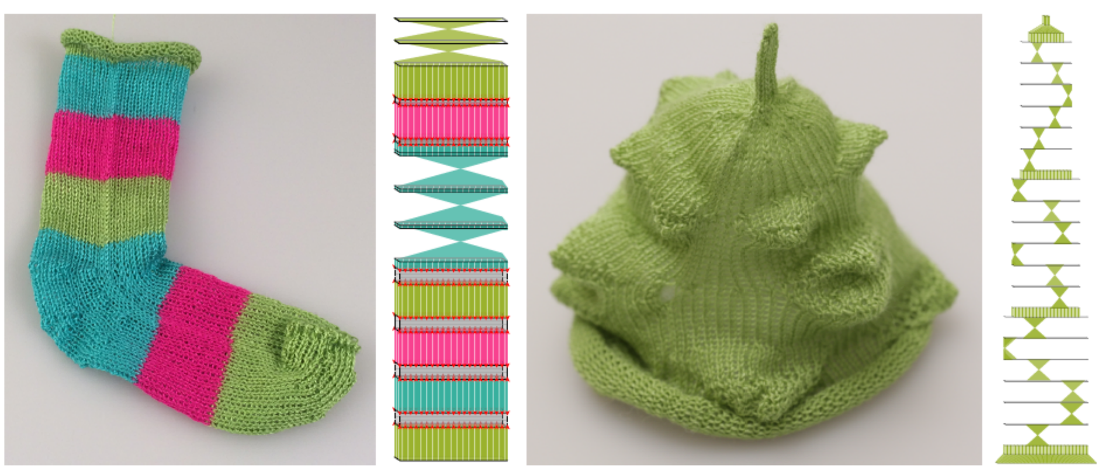


Figure 3.20: Partially-knit rows can be used to bend shapes, as in the heel of this sock, and create bulges, as in this whimsical hat.

Shapes can also be modified by knitting partial courses, called short rows. These short rows have the effect of pushing their adjacent courses apart, creating a bend or bulge in the fabric (Figure 3.20). An example is the traditional shaping of a sock, where extra rows are added to create the rounded part of the heel.

3.3.3 Compiler Pipeline

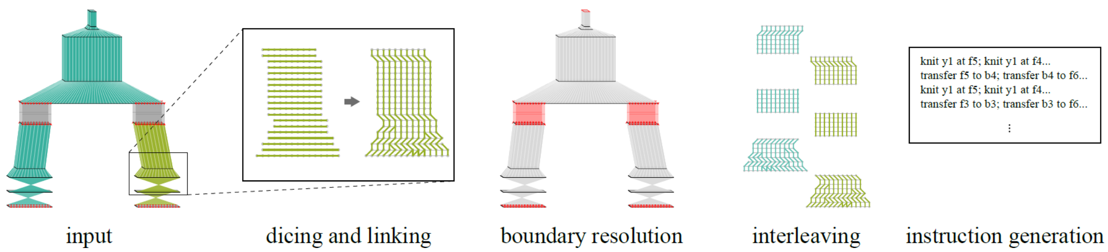


Figure 3.21: Our compiler pipeline. Our compiler first dices each of its input primitives into courses, assigns short rows to be knit between their adjacent courses, and decides how to link stitches in adjacent courses together. Next, during the boundary resolution stage, it decides how to start and end each primitive. Finally, it interleaves the knitting and linking steps required for all the primitives into a final ordering, and generates knitting assembly language instructions for them.

Our compiler transforms high-level primitives into knitting assembly language commands through a series of stages illustrated in Figure 3.21. Given a group of input primitives to knit, our compiler first dices each primitive into courses (assigning short rows to be knit at the same time as their adjacent course), and then determines how they should be linked by respecting the stretch limitations of the yarn and evaluating how to match the size of the adjacent courses. Next, during boundary resolution, the compiler selects operations to perform at the start and end of each primitive, based on definitions specified in the input and the needle occupancy of the diced primitives. Finally, linking and knitting steps are interleaved based on their construction time, with stash/unstash blocks added as needed to ensure that no yarn is stressed during linking, and knitting assembly language instructions are emitted.

Input

Most objects created on a knitting machine, such as gloves, sweaters, stuffed toys, and socks, are assemblies of tubes and sheets, with varying radii and bends. Motivated by this observation, we designed our compiler to take as input a list of tube and sheet primitives, positioned according to their construction time and location on the knitting machine bed (e.g., Figure 3.6 and 3.20). This layout makes it easy to ensure that no two primitives require the same needles at the same time, because this would result in an overlap in the input.

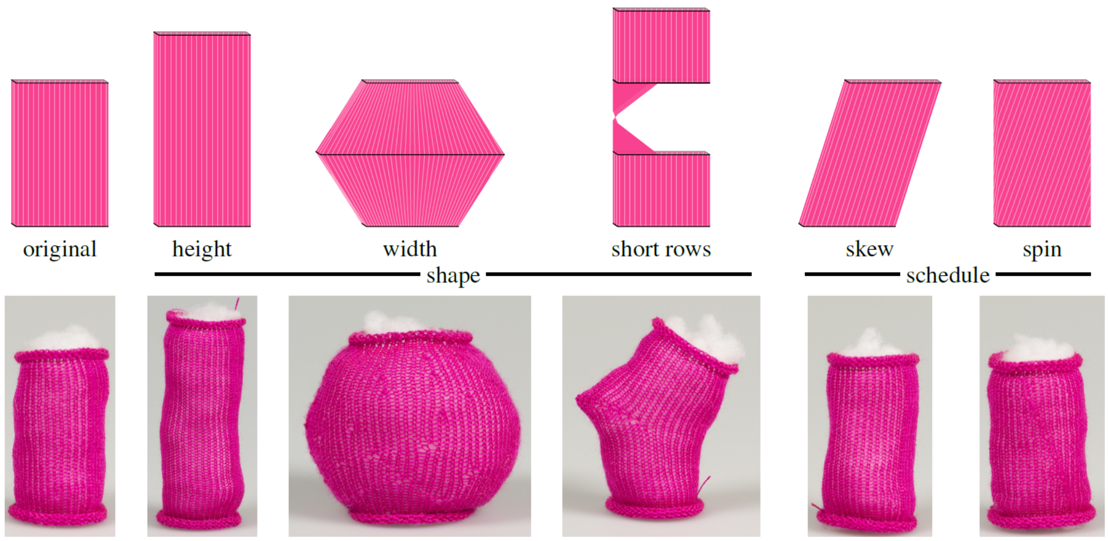


Figure 3.22: The degrees of freedom of a tube in our input format. Schedule parameters do not change the final shape.

Primitives have both shaping parameters, which influence their final appearance, and scheduling parameters, which change which needles are used to construct them (Figure 3.22). For tubes, the shaping parameters are: **height**, the number of courses; **circumference**, the number of stitches in each course; and **short rows**, partial courses that will cause the tube to bend. The scheduling parameters map each course to the needle bed: **time** indicates when the first course will be knit; **skew** gives the horizontal position of each course; and spin rotates courses on the bed. In our compiler's input file format, all parameters except for **height** and **time** are linearly interpolated along the height of the tube.

Sheets are specified relative to a "supporting tube" and have one additional shape parameter: the **percent** of the supporting tube's circumference occupied by the sheet. For example, a sheet at 100% is a tube with a slit in it (useful as a thumb slit when making a hand warmer, Figure 3.33).

All dimensions are specified in (possibly fractional) numbers of stitches, that, given a yarn, can be converted to real-world lengths. Thus, porting a design from one yarn to another requires only that one multiply all dimensions by a constant factor determined by the yarns' relative stitch size.

Dicing

After reading input primitives, our compiler breaks them into horizontal slices (courses). Computing these needle lists from the supplied input parameters is straightforward rasterization. Each course has an abstract "knitting time" value, a list (in counterclockwise order) of needles where loops will be formed, and a parameter value for every needle (running from 0 to 1 counterclockwise around the cycle) that will be used in linking. Short rows are treated as part of their closest course.

Linking

Once a primitive has been transformed into courses, links between adjacent courses are made. These links will be used to generate transfer instructions later. Linking is accomplished by selecting an optimal combination of 1-1, 2-1 (decrease), and 1-2 (increase) links between stitches in adjacent courses, in a process akin to stretch-limited dynamic time warping. The objective function to be minimized is the sum of absolute differences between the parameter value at each stitch and the parameter value at its assigned location, plus a large additional penalty for every increase or decrease. This additional penalty prevents the optimization from arbitrarily introducing paired increases and decreases.

Using a parameter value, rather than simply linking stitches that are closest on the bed, is the key to separating scheduling (needle location) and shape (stitch connectivity). Without using parameter values to accomplish linking, it would be impossible to spin or skew primitives on the bed without also distorting them.

Boundary Resolution

Once all the primitives have been diced and linked, their first and last courses are compared in a stage we call boundary resolution. As we have belabored, knit items are created by pulling loops through loops. Any loop not pulled through another loop could unravel, causing the final item to fall apart. Thus the starting and ending loops of each primitive must be made stable by being pulled through or pulling through another loop. This stabilization can be accomplished by attaching the loops to loops from another primitive, resulting in the primitives being attached ("glued"); alternatively, local techniques can be used to stably create ("cast on") or stabilize then drop ("bind off") loops.

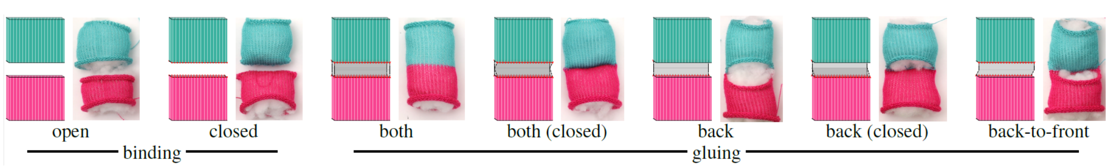


Figure 3.23: In our compiler's input, each primitive has a start and end boundary definition that indicates how to stabilize loops. These can result in primitives that are open, closed, or attached together in various ways. Not shown are front, front (closed), and front-to-back gluing, which are defined analogously to their back variants.

Each input primitive includes a *boundary definition* that describes which of these actions to take at its start and end. Our compiler supports various binding and gluing styles, illustrated in Figure 3.23. Boundaries are the one place in the primitive definition where the scheduling parameters, specifically, spin, can influence structure, as shown in Figure 3.24.

Figure 3.24: Primitive scheduling (particularly, spin âĂŞ the orientation on the bed) is important at boundaries. This tube has closed boundaries at the top and bottom, but has had its "spin" scheduling parameter adjusted at the top boundary.

Interleaving and Instruction Generation

Once the boundaries have been marked, courses and links between courses are sorted based on their knitting time. The knitting time for links is set to halfway between their adjacent courses. The compiler walks through this sorted list, tracking which needles are currently in use and emitting instructions.

For links, the compiler calls on our transfer planning algorithm (section 3.3.4, Transfer Planning) to generate a series of transfer instructions to enact the link. If the generated plan involves any large racking, it may stretch or break *other* primitives currently held on the bed (Figure 3.25). To avoid this, the compiler will prefix the transfer instructions from the link with what we call a *stash* operation, which moves every primitive currently attached to two beds onto one bed, using

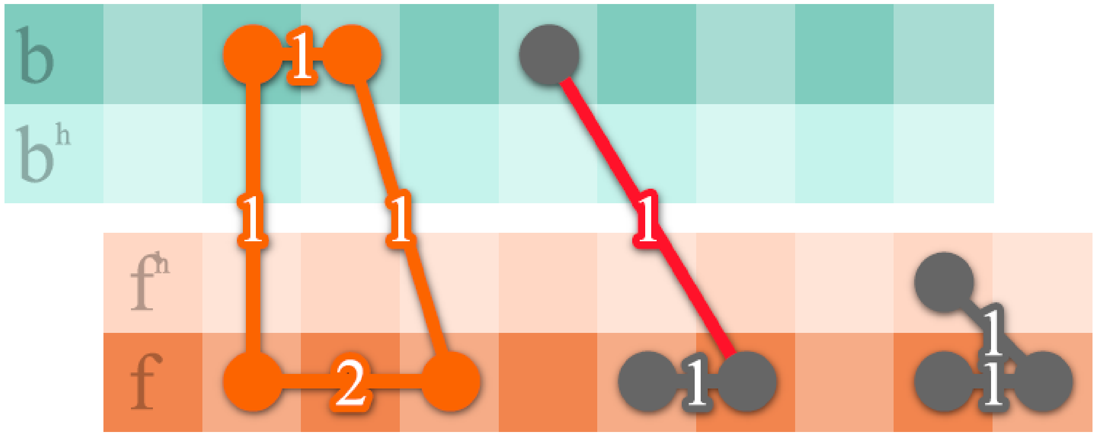


Figure 3.25: As part of moving a target cycle (left), our transfer planning algorithm may generate rackings that stress other primitives (center). In this case, our compiler will "stash" the other primitives on one bed by using the holding hooks (right).

the holding hooks.

For courses, the compiler first emits transfer instructions to unstash any required stitches that were previously stored on holding hooks, and then walks through the course, emitting a knit instruction for every needle in the course. In the case of needles that do not currently contain a loop because of the previous link, our compiler follows the common knitting practice of using a split or tuck instruction to fill the gap. If the course is a boundary, instructions are emitted to enact the requested binding or glue operations.

Backend

We have developed a translator from our knit assembly language to the '.dat' format used by Shima Seiki's knit specification system, KnitPaint. We use KnitPaint to translate these files into Shima Seiki's proprietary machine control format.

Though our compiler assumes an x-bed machine, we only have access to a v-bed machine (no holding hooks); thus, as part of this translation process, our compiler transforms x-bed instructions into *half-gauge* v-bed instructions, where alternate needles are used to emulate holding hooks (Figure 3.26:

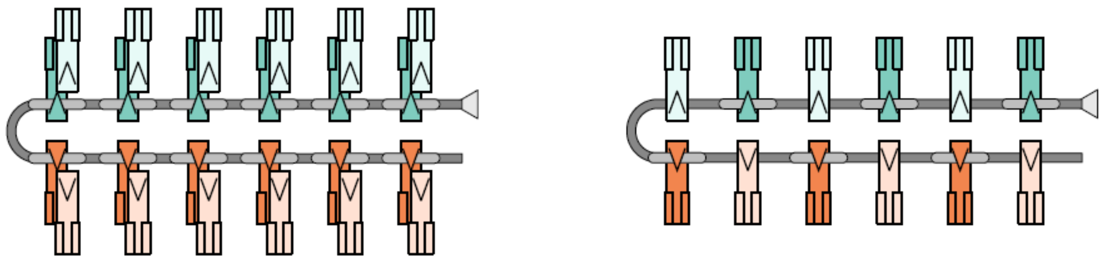


Figure 3.26: The top view on the left is the same holding hooks on the x-bed machine as figure KL. The top view on the right is our v-bed machine using alternating hooks as holding hooks (the dark-colored variants being the holding hooks). These 'holding hooks' do not block normal operations as an x-bed's holding hooks, but we treat them the same.

Our translator also handles low-level tasks such as inferring yarn carrier motion from knitting instructions, unifying instructions into "passes," generating machine-specific boilerplate for yarn insertion and removal, setting racking flags appropriately, setting yarn tensions, and performing appropriate cast-on and bind-off stitches.

Given that knitting assembly language operations all correspond to generally-available machine capabilities, there is no technical reason that a similar program could not be developed to output, e.g., Stoll's SINTRAL machine control language.

3.3.4 Transfer Planning

So far, this project has described providing macros that combine and translate machine instructions into more comprehensive knitting commands. While these macros help efficiency and clarity in working with machine knitting design software, they do not provide the needed safety for users to confidently use the design tool. Yarn only has a certain amount of tensile strength, which restricts how far stitches can be stretched between needles and beds. Not respecting this limitation will cause the needles to bend or break as the machine attempts to perform operations that the common materials are unable to support. In order to respect the constraints required by generating general links between courses, our compiler needs a way of generating transfer instructions to safely move collections of stitches around on the needle beds. These safe transfer instructions, if done properly, also require dozens or hundreds of low-level operations that should be abstracted for the user for additional ease and safety.

During knitting, our primitives occupy needles arranged in *cycles* (counter-clockwise loops), constrained to obey a given *slack* (distance between stitches).

Definition 1 (Cycle). Define a *cycle* to be a collection of needles $\mathtt{c} = [c_1, ..., c_n]$ that, when connected in index order, form a non-self-intersecting loop on the needle bed. When dealing with cycles, we will use cyclic indexing (so we define $c_{i+xn} \equiv c_i$ for integers x).

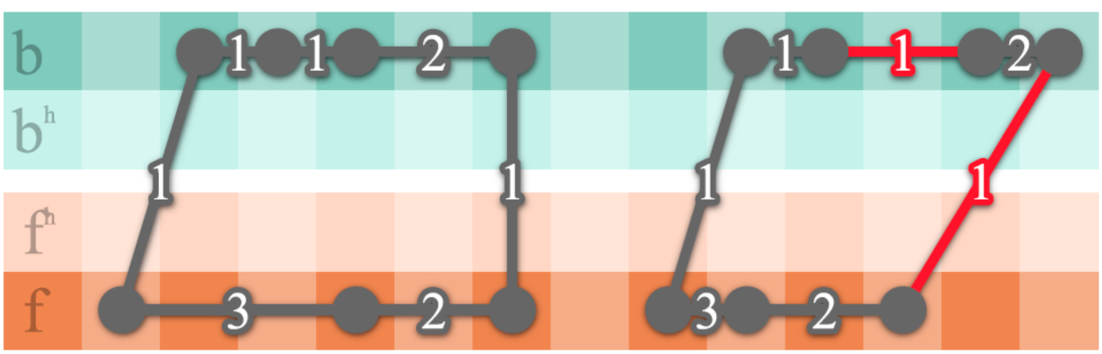


Figure 3.27: Cycles and slack. When drawing a cycle, we typeset slack as labels on edges. Both cycles above have the same slack, but the one on the left respects that slack, while the one on the right places some stitches too far apart (red edges).

In our case, cycles are commonly the ends of tubes that have been previously knit. It is important not to excessively stretch the yarn between adjacent stitches in the cycle, so we introduce the notion of *slack* (Figure 3.27):

Definition 2 (Slack). Define *slack* to be a list of non-negative integers $\mathtt{s} =$

$[s_1, ..., s_n]$ that indicates the greatest allowed length between subsequent locations in a (cyclic) list of needles. We say a list of needles a *respects* slack $\mathtt{s}$ if $\forall i$: $|a_{i+1} - a_i| \leq s_i$ (notice that, by cyclic indexing convention, s_n is the slack between the first and last needle).

With these definitions in hand, we may finally define the transfer planning problem:

Definition 3. An instance of the **transfer planning problem** consists of a slack $\mathtt{s}$, counterclockwise cycles of needles $\mathtt{n}$ (the source) and $\mathtt{n}'$ (the target) a maximum racking $R \geq 1$, and a free range $[min, max]$ such that $|\mathtt{s}| = |\mathtt{n}| = |\mathtt{n}'|$, both $\mathtt{n}$ and $\mathtt{n}'$ respect slack $\mathtt{s}$, no needles are split (i.e. $n_i = n_j \implies n'_i = n'_j$), slack is at least one between all needles, and all needles in the source and target are contained in $[min + 1, max - 1]$.

A list of transfers is a **solution to this instance** if it transforms $\mathtt{n}$ to $\mathtt{n}'$, requires no racking greater than R, and all intermediate poses lie in free range $[min, max]$ and respect slack $\mathtt{s}$.

Our compiler is based on the first — to our knowledge — general solution to this problem. Our solution is heuristic but complete; that is, it can solve all instances of the problem, but it may use more transfers than necessary. We define the algorithm here and sketch a proof of completeness.

The outline of our algorithm is as follows: we define a penalty function that takes positive integer values for all bed configurations except the target configuration $\mathtt{n}'$ (at which it is zero) along with a type of transformation (collapse/expand); our transfer planning algorithm will compute two locally optimal collapse/expand transformations at each step and choose whichever decreases the penalty the most.

The Roll-Goal Penalty

For transfer planning, we need a penalty function that measures the distance from the current configuration to $\mathtt{n}$, does not have any local minima, and can be computed as a sum over penalties at each needle. (This last condition makes computing the locally optimal behavior with dynamic programming feasible.)

We decided to use a penalty involving the distance to the goal needle measured *around* the bed. That is, for each stitch n_i in the cycle, we assign a roll number r_i indicating the number of times that stitch should switch beds to the left (if $r_i < 0$) or right (if $r_i > 0$) of the other stitches.

Given an assignment of roll numbers, it is straightforward to define a penalty

$$\text{penalty}(\mathtt{n,\ r,\ n'}) \equiv \sum_i \mathrm{p}(n_i, r_i, n_i') \tag{3.10}$$

where

$$\mathrm{p}(n_i, r_i, n_i') \equiv \begin{cases} |n' - n| & \text{if} r = 0 \\ 2 + (n - min) + \mathrm{p}(min, -(r+1), n' & \text{if} r < 0 \\ 2 + (max - n) + \mathrm{p}(max, -(r-1), n' & \text{if} r > 0 \end{cases} \tag{3.11}$$

We illustrate the individual-stitch penalty $\mathtt{p}$, which measures distance "around" the free needle range, in Figure 3.28.

It remains to assign roll numbers. We will define these in terms of winding numbers w_i, whose values will indicate how many times a stitch should be rolled counterclockwise.

Defining δ_i to be the number of times the cycle switches beds in counterclockwise order between n_i and n_{i+1}, and δ_i'i analogously for $\mathtt{n}'$, it must be the case

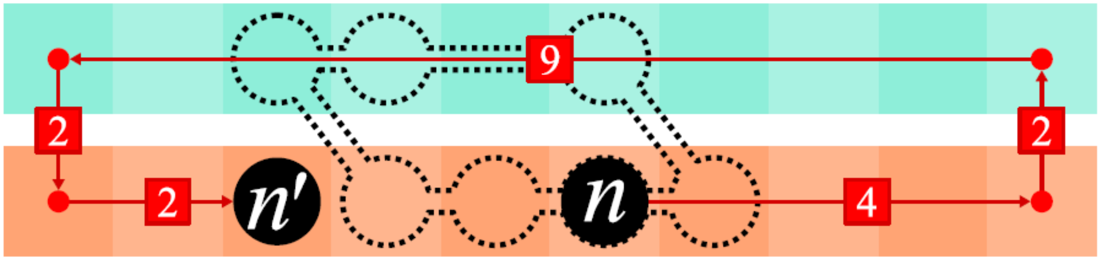


Figure 3.28: Visualizing the penalty $\mathtt{p}(n, 2, n') = 19$ computed for stitch n with goal n' and roll number $+2$. The penalty is computed by walking around free needle range in the direction indicated by the roll number, charging 1 for every needle traversed and 2 for every change of bed.

that

$$w_i - \delta_i = \delta_i' = w_{i+1} \tag{3.12}$$

This equation determines w_i, up to global addition of an even number.

The conversion between roll and winding numbers is straightforward:

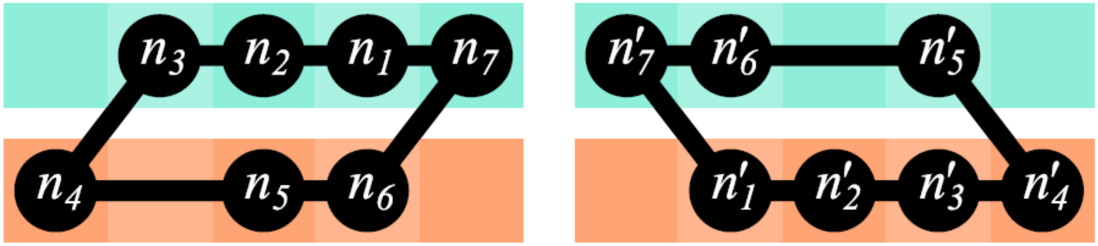
$$r_i = \begin{cases} -w_i & \text{if } n_i \text{ is on the back bed} \\ w_i & \text{if } n_i \text{ is on the back bed} \end{cases} \tag{3.13}$$

An illustration of winding number assignment is shown in Figure 3.29.

Collapse-Expand

Our transfer planning algorithm uses a sequence of collapse-expand transformations. Each collapse-expand transformation (Figure 3.30) collapses the cycle onto one bed, moves it to the other bed, and expands it back to both beds, with each motion providing an opportunity to reposition stitches in order to reduce the overall penalty.

In the following description and accompanying figures, we show a collapse-expand from the back bed to the front bed; however, it is equally reasonable to



i	1	2	3	4	5	6	7
δ_i	0	0	1	0	0	1	0
δ'_i	0	0	0	1	0	0	1
w_i	1	1	1	0	1	1	0
r_i	-1	-1	-1	0	1	1	0

Figure 3.29: Winding (w_i) and roll (r_i) numbers are determined by where cycles `n` and `n`$'$ cross between the front and back bed.

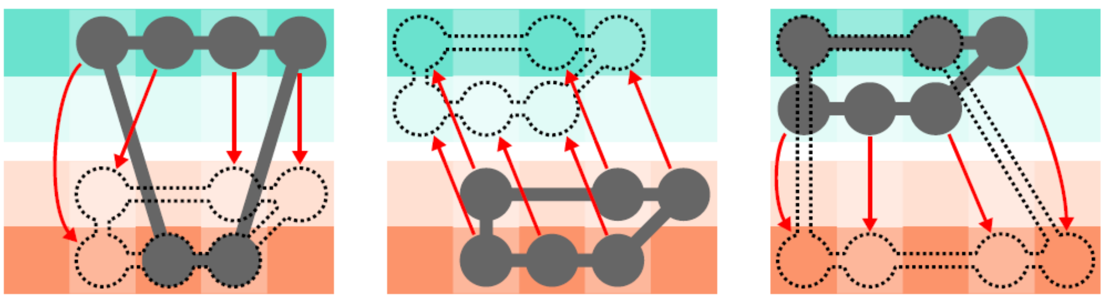


Figure 3.30: During a collapse-expand transform, back-bed stitches are collapsed to the front bed, all stitches are moved to the back bed, then the cycle is expanded by moving some stitches from the back bed to the front bed.

switch the roles of the beds, and our overall planning algorithm considers both options.

The collapse phase will, at each step, either transfer the leftmost or rightmost stitch remaining on the back bed to the hooks or needles of the front bed. This ordering ensures that there are at most two places where the slack between adjacent stitches must be respected.

After the collapse phase, the cycle is moved from the front bed to the back bed to allow the expand phase to access the front stitches. The algorithm will shift the whole cycle by a constant offset if doing so decreases the penalty function while not moving any stitches outside the free needle range.

The expand phase moves the stitches on the holding hooks of the back bed to the front bed. Expand phases used by our algorithm can start with any stitch, but proceed by transferring only stitches adjacent to those already transferred. This constraint again ensures that there are at most two places that must be checked for valid slack.

An optimal collapse or expand for a given state may be computed with a memorized recursive function in $O(|\mathbf{n}|^2 R^2)$ time. This complexity bound arises from the fact that the actions available during a collapse or expand depend only on the indices of the next-to-be-transferred stitches and the locations of the previous stitches.

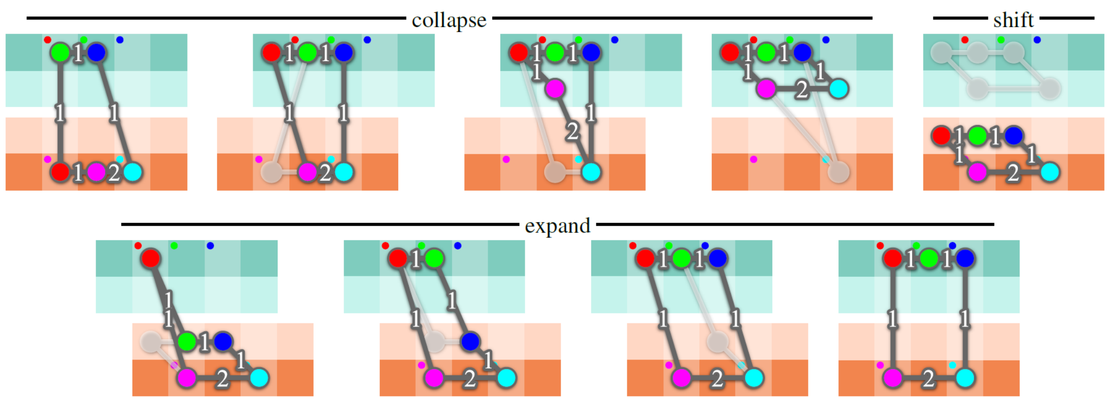


Figure 3.31: A transfer plan generated by our algorithm. Solid colored circles indicate goals. First, **top**, the collapse phase moves stitches to the needles and hooks of the back bed. Next, **top right**, the collapsed cycle is moved to the front bed. Finally, **bottom**, the expand phase moves stitches from the front bed back to the back bed.

Our planner proceeds by repeatedly choosing an optimal collapse followed by an optimal expand; one such pair is shown in Figure 3.31. It is worth noting that this is not the same as an optimal collapse-expand *pair*; however, choosing separately avoids the prohibitively high complexity of a full search, and it is sufficient for correctness.

Final Algorithm

Our final transfer planning algorithm (Figure 3.32) first assigns roll values to every stitch and then iteratively makes a greedy choice of collapsing/expanding to the back or front bed.

Algorithm 1 Transfer planning outer loop

procedure TRANSFERPLAN($\mathbf{n}, \mathbf{n}', \mathbf{s}$)
    $\mathbf{c} \leftarrow \mathbf{n}$
    $\mathbf{r} \leftarrow \text{AssignRoll}(\mathbf{n}, \mathbf{n}')$
    $T \leftarrow []$
    while $\text{penalty}(\mathbf{c}, \mathbf{r}, \mathbf{n}) > 0$ **do**
        $(T_\mathrm{f}, \mathbf{c}'_\mathrm{f}, \mathbf{r}'_\mathrm{f}) \leftarrow \text{CollapseExpand}(\mathbf{Front}, \mathbf{c}, \mathbf{r}, \mathbf{n}', \mathbf{s})$
        $(T_\mathrm{b}, \mathbf{c}'_\mathrm{b}, \mathbf{r}'_\mathrm{b}) \leftarrow \text{CollapseExpand}(\mathbf{Back}, \mathbf{c}, \mathbf{r}, \mathbf{n}', \mathbf{s})$
        if $\text{penalty}(\mathbf{c}'_\mathrm{f}, \mathbf{r}'_\mathrm{f}, \mathbf{n}') \leq \text{penalty}(\mathbf{c}'_\mathrm{b}, \mathbf{r}'_\mathrm{b}, \mathbf{n}')$ **then**
            $(\mathbf{c}, \mathbf{r}) \leftarrow (\mathbf{c}'_\mathrm{f}, \mathbf{r}'_\mathrm{f})$
            $T \leftarrow \text{cat}(T, T_\mathrm{f})$
        else
            $(\mathbf{c}, \mathbf{r}) \leftarrow (\mathbf{c}'_\mathrm{b}, \mathbf{r}'_\mathrm{b})$
            $T \leftarrow \text{cat}(T, T_\mathrm{b})$
        end if
    end while
    return T
end procedure

Figure 3.32: The transfer planning outer loop algorithm.

3.3.5 3D Knitting Results

All knit objects in this section were created with our compiler and knit on a Shima Seiki SWG091N2 15-gauge v-bed knitting machine. This is a two-bed machine with 15 needles per inch and a 91cm long needle bed. It has a maximum racking value of eight needles and includes ten yarn carriers. We knit our designs in half-gauge (section 3.3.3), as this machine is not an x-bed machine. Knitting at half gauge results in a somewhat loose knit, because the 15-gauge needles cannot hold yarns thick enough to produce a dense 7-gauge knit.

The machine is able to use a wide variety of yarns, though for this project our examples were knit in acrylic ("Supersheen 1-ply" from Yeoman Yarns) and merino wool ("Polo 1-ply" from Yeoman Yarns). Note that Yeoman uses "ply" in the UK sense: as a size designation that does not reflect the number of component yarns plied together to form these yarns. On our machine at half gauge, these yarns both produce stitches that are approximately 3.04mm wide by 1.58mm tall; values we obtained by knitting a tube, stuffing it, and computing the circumference and height of a known stitch-count portion.

Compile time is generally trivial compared to knitting time, with the most expensive steps of compilation being the linking and transfer planning stages. This observation is unsurprising; both are worst-case cubic in the number of stitches.

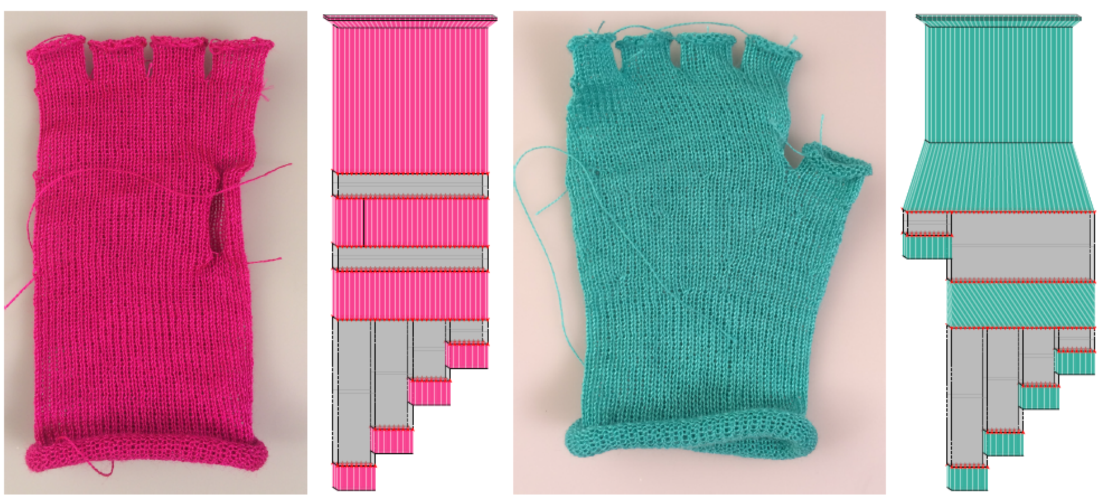


Figure 3.33: Two hand-warmers designed in our system. One uses a sheet to create a slit for the thumb, while the other uses another tube for the thumb then decreases the width of the main tube to fit the wrist.

We used our system to create clothing objects for plush toys (Figure 3.6) and people (Figure 3.33). Having primitives with easy-to-adjust size values is important in both cases. Our sock example (Figure 3.20) is far too small to be worn, but does demonstrate how gluing together tubes knit with different yarns can make colorful stripes.

We also created a number of knit toys. Our snake toy (Figure 3.34) shows how easy it is to build higher level primitives — its body is the result of writing a script to translate a helix into a tube with short rows. The Hilbert curve (Figure 3.35) was generated by a similar script. Our collection of robots (Figure 3.36) demonstrates the benefit of high-level primitives for rapid iteration. Finally, the teapot example (Figure 3.37) makes heavy use of skew operations to schedule its spout and handle; indeed, some of the stitch motions in this example are so long that they do not consistently knit; in the version we show, several dropped stitches were manually corrected.

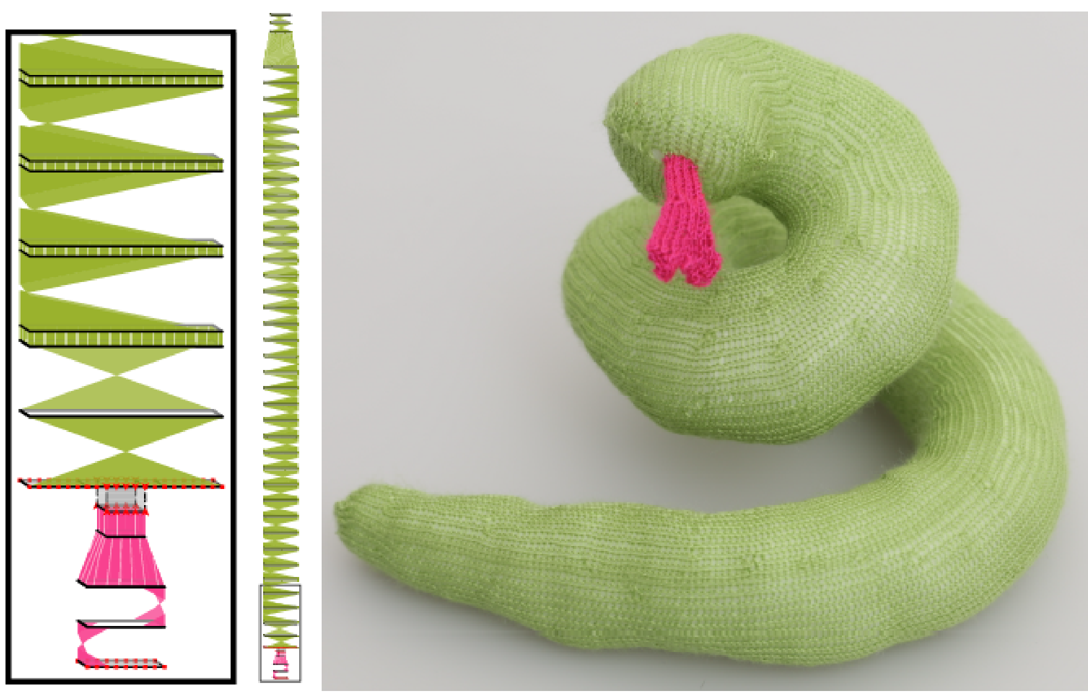


Figure 3.34: The helical shape of this snake is the result of many sets of short rows.

Figure 3.35: These plush robots are all variations on a design, created rapidly by editing high-level primitives to be smaller, with a chunkier torso and claws, and in a seated posture.

Graphical Editing

We developed a graphical interface to edit our compiler's input format. The user interface (Figure 3.38) consists of linked preview and bed views. The preview view shows a rough approximation of the final 3D shape of the model, computed in a simplified way (without consideration for knit fabric physics) using as-rigid-as-possible alignment of tube primitives. The bed view shows the input to the compiler, and provides basic vector-graphics-style editing capabilities for primitive positioning and sizing, along with some special tools to handle the spin and skew degrees of freedom, adjust the width of sheets, and set boundary definitions. As a convenience, the interface also loads a description of the color information of our yarns, and can use this to set the color of the shape primitives.

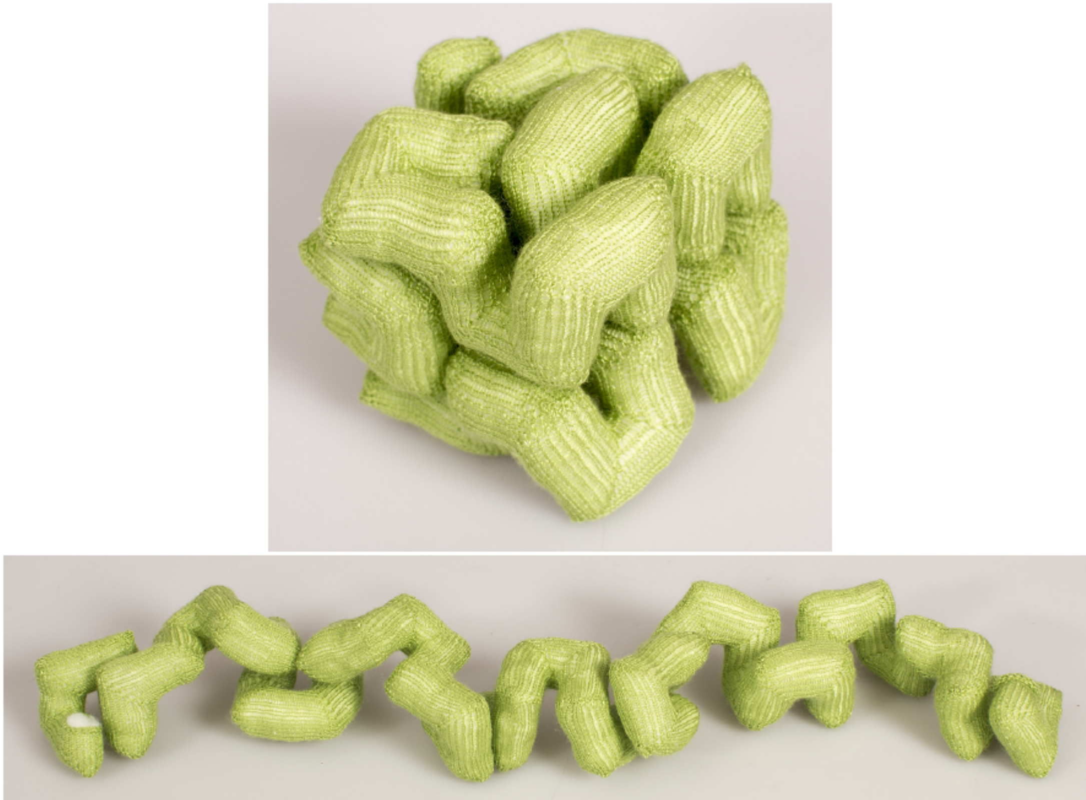


Figure 3.36: Two views of a 3D Hilbert curve of order two, generated by using a small script to write an input file for our compiler.

3.3.6 3D Knitting Discussion

We introduced a compiler which can translate high-level shape primitives into low-level knitting instructions. The core of our compiler is a transfer planning algorithm for knit cycles, which is *provably correct*[4] on all inputs, though it may not produce time-optimal plans. In addition, we presented a formal, general treatment of the basic operations of a knitting machine; this “knitting assembly language” could be further refined into a general file format for describing knitting — both for knitting machines, and perhaps even for rendering and simulation algorithms.

[4]See the supplementary materials for the original publication for the full proof [124].

Figure 3.37: This teapot makes extensive use of the "skew" scheduling primitive.

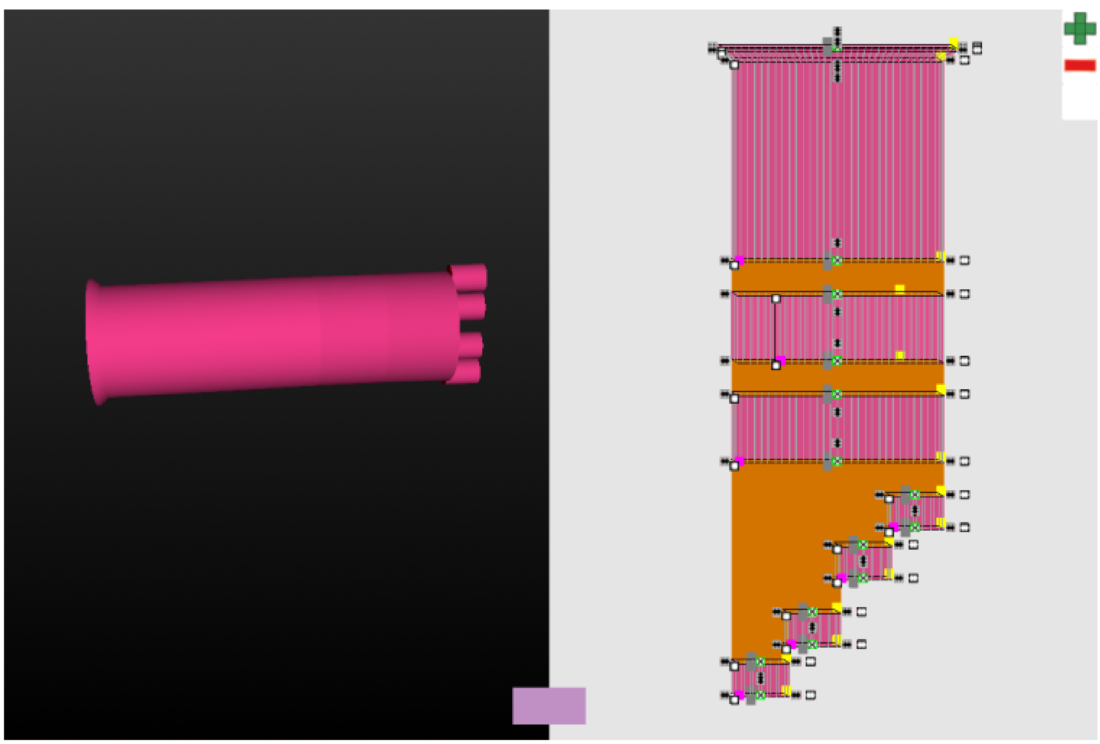


Figure 3.38: The 3D preview (left) and 2D bed view (right) of our interface. The displayed object is the left hand-warmer from Figure 3.34.

The focus of our compiler is to make it easier to create *knit structures*; however, many knit objects also use various combinations of stitches to create interesting surface *textures*. In the future, we plan to extend our compiler to support surface texturing, by adding local texturing programs akin to fragment shaders. These shaders will be constrained to not change global structure, a process which is complicated by the prevalence in knitting of mid-scale structures (e.g., “cables”) which use the local movement of small blocks of stitches to create texture.

We believe that the knit fabrication community can benefit from a uniform control language akin to our knitting assembly language, and it is part of our ongoing research to port the language to any knitting machines that we are able to access. Such a language also allows for uniform structural debugging tools and error checking (e.g., “you stretched that loop out a long way; it might break”). It would also be interesting to create a knitting assembly language backend that could set up yarns for simulation (e.g., by creating a stitch mesh [219]); this backend would allow physical and virtual garments to be created using the same code.

Our pipeline presents many opportunities for small refinements: during dicing, it would be interesting to try more sophisticated rasterization techniques (possibly involving “hinting” akin to that used in font rendering); linking could be user-controlled for interesting shaping effects; transfer plans could be optimized further, possibly by using limited lookahead in the collapse-expand planning space; and, with some reverse engineering work, the backend could write machine control files directly.

Our transfer planning algorithm cannot work with general (i.e., non-cyclic) slack constraints. Indeed, transfer plans do not always *exist* in the presence of general yarn constraints — consider a cycle with an additional edge linking the front and back beds at its center; this cycle cannot be rotated 180 degrees even though both the starting and ending positions are valid. It would be interesting to attempt to characterize the space of feasible transfer planning algorithms.

Our compiler uses unoptimized implementations of transfer planning, which can approach $O(s^3)$ for pessimal inputs; transfer planning takes $O(s^2)$ per step, and may require $O(s)$ steps to complete a plan). Both of these could likely be sped up by an order of magnitude by using early-out checks (e.g., the collapse phase could terminate as soon as it knows it does not have space to place all remaining stitches; the time warping phase could not consider paths that cannot possibly align based on the slope constraints), and, further, are independent per-course tasks, so could be distributed across parallelized software threads relatively easily.

The knitting machine we used for output, like many machines, actuates its needles using a cam system that slides along the beds. This design means that, in practice, it takes basically the same amount of time for the machine to operate on *any* number of needles, as long as those operations can all be performed during one *pass* of the cam system. (In addition, some operations can only be performed when the pass is being made in a specific direction.) A production-level system would almost certainly want to carefully track machine state for the current target machine and attempt to use production time when breaking ties between otherwise equivalent actions.

Right now, consumer-level knitting machines lack the sophisticated transfer capabilities of industrial machines — indeed, the basic home knitting machine has not changed in hardware capability since the 1980s. It might be interesting to come up with optimization passes for our compiler that could make it as easy as possible to construct objects on these severely restricted machines. There is also a vibrant community developing new technology for these machines [139, 79, 11], so perhaps more automated home knitting is within reach of domestic crafters, if not amateurs. Until that time, having a common instruction format (e.g., our knitting assembly language) could make it much easier to send knit jobs to a central location for industrial machine processing.

We believe that 3D machine knitting should join 3D printing in the pantheon of end-user-accessible additive fabrication, and that getting it there will require new tools, algorithms, and data exchange formats, of which our compiler, transfer planning algorithm, and knitting assembly languages are first examples.

3.4 Textile Craft Ideation and Design Insights

3.4.1 Domain-Specific Languages

Both projects in this chapter chose limited domains: a sub-genre/style of embroidery, blackwork, and knitting as made possible on industrial knitting machines and their low-level machine operations. The high-level abstractions both of these projects accomplished made using their tools more efficient, easier, and in some cases safer than using low-level operations, which makes room for ludic engagement with the domain. Abstractions help make the representation of digital design spaces closer to that of non-digital spaces, which help bridge the gap between physical and digital agency. These limited domains could also be assured to be valid, accurate, and able to be manufactured in the real world, within reason.

A systems-level understanding of the domain's possibility space was required. In both cases, we surveyed as many examples of output as we could find. We cataloged techniques seen in those samples, as well as experimented with those of our own design to confirm our understanding. For example, the tuck increase in Figure 3.19 that left a hole, which became even more pronounced when stuffed, was never preferred compared to a split increase, as structural integrity of the final design was of utmost importance. Techniques that were seen as fringe, optional, or statistically uncommon were isolated and given a lower priority for more important techniques. For example, cables are a common knitting operation by hand, but the knitting algorithm would have been much more difficult to account for the extra needle allocation to hold and swap stitches.

In the case of blackwork embroidery, the designs were intended to be embroidered by a machine, which has difficulty making jumps across gaps in unconnected designs. We discouraged making isolated designs by making them less likely and more difficult to execute in the user interface. Certain techniques, such as shading a design via removing more and more lines over a given distance (Figure 3.39), were not supported given how many gaps would be created given arbitrary removal of stitches. Had we focused on hand embroidery rather than machine, these features would have been much easier to execute and interesting to experiment with.

Difficulties: A Crochet Example

What happens if we pick a challenging domain, or do not account for its variability? There are no proper industrial crochet machines that can mass-produce crochet fabric as there are knitting machines [100, 173]. The structure of crochet offers only 1 general point of contact to the whole fabric (yarn on the needle) and the rest floats, twists, or is held by the crocheter. Crochet stitches are also more complex than knitting machines, with 5 standard stitches and hundreds of blends, which can be attached to any point on the existing fabric or the hook (Figure 3.40).

Suffice it to say, while knitting has been formalized — by our project, and by machines that include techniques such as lace and cables — crochet has not been

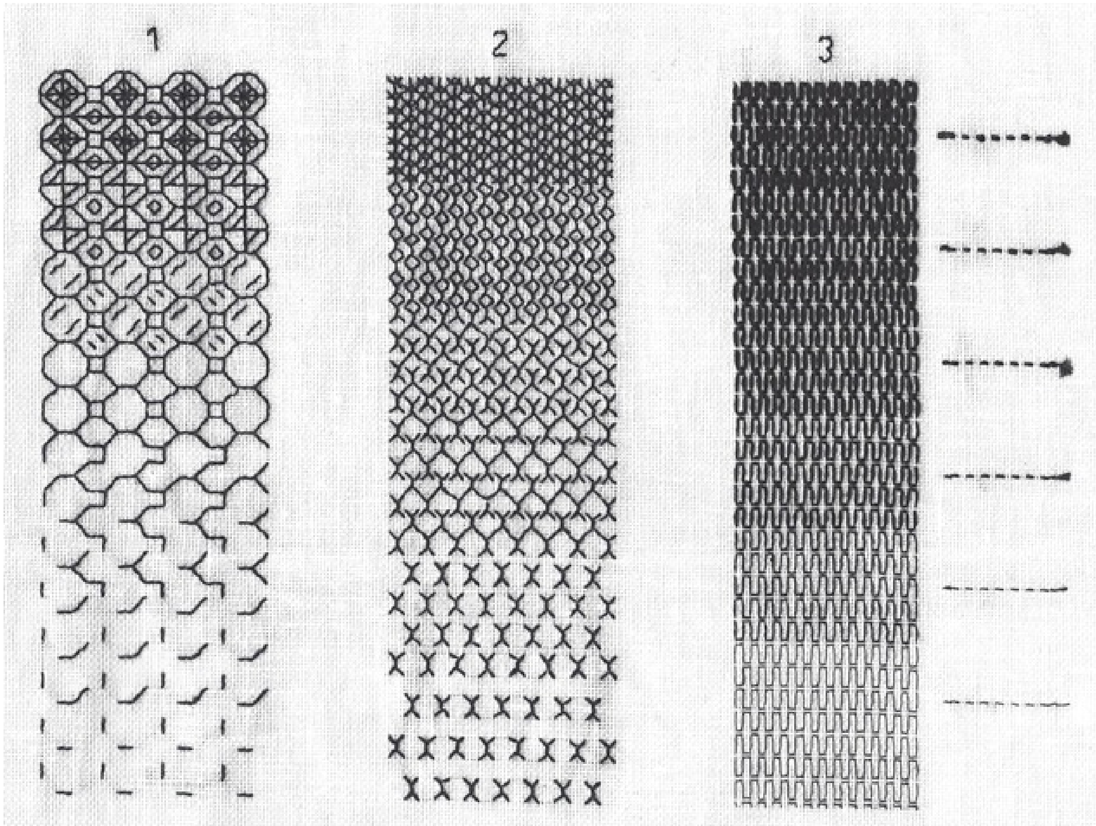


Figure 3.39: An example [93] of blackwork embroidery gradients, a feature specifically not included in the embroidery project for its tendency toward gaps (some versions of this approach reduce down to single stitches). The image shows three examples of gradient shading, where (1) and (2) are shaded using density of stitches, while (3) shows a gradient via the thickness of the thread.

formalized to the point of automation. Sub-domains must be defined if software or manufacturing machines are to have any hope of aiding humans meaningfully in the huge design space of crochet. Without high-levels of abstraction, digital tools lose their advantage over non-digital design spaces, and any potential benefit of digital agency is lost.

A Tiny Crochet Pattern Grammar

For a more practical example, one might make a simple grammar meant to make one flat row of crochet shown in this section. Figure 3.41 describes the main standard crochet stitches and their notation that will be used in making a pattern

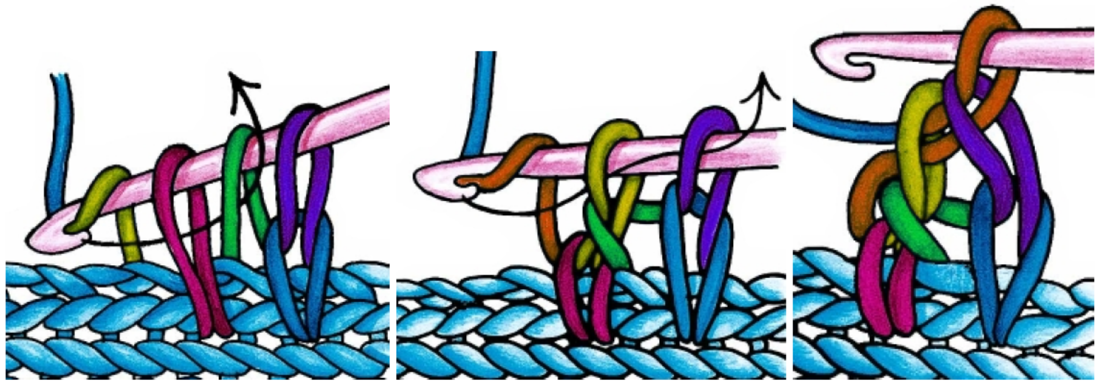


Figure 3.40: The end snapshots of a double crochet stitch progression after 2 chain stitches (the two loops on top of each other, dark blue and purple, on the far right side) to start the row. The dark purple loop acts as the one point of contact before the stitch began and will act as the top of the double crochet stitch once it is completed. Each colored section of yarn is a different loop that makes up this one stitch. Excluding the dark blue chain stitch which is technically not a part of the double crochet, there are five interconnected loops to this stitch. Knitting stitches only have one loop, whether it be knit or purl.

20 stitches long.

Similar to the blackwork embroidery approach in section 3.2, small groups of stitches are generated and arranged/mirrored/repeated to make rows of crochet with different stitches in Figures 3.42 and 3.43.

What would happen if we wanted to expand this grammar to make multiple rows of crochet? Unlike knitting, these rows have different kinds of topology based on the existing stitches and how they are arranged. Not only can new stitches be inserted in, around, and between each of these existing stitches, but if the topology is not respected, the fabric will no longer be flat or rectangular in shape and may no longer be appealing or suit the designer's goals.

3.4.2 Designing Design Software Interfaces

The priorities of both the embroidery and knitting projects presented in this chapter were the abstracted models of their individual domains. The knitting

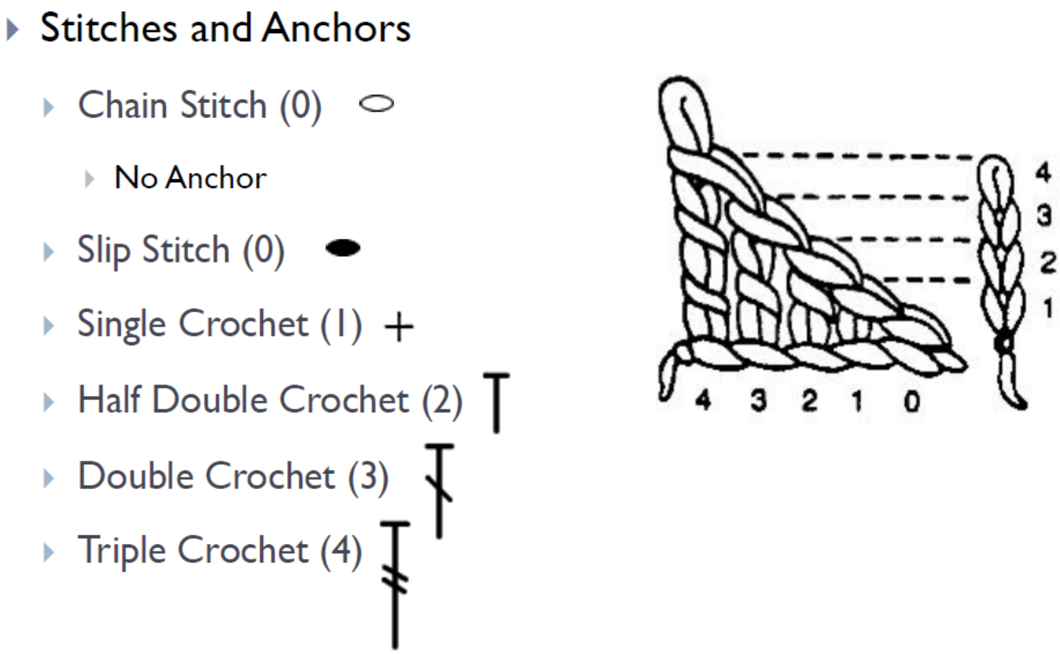


Figure 3.41: Crochet notation and height definitions of foundation stitches and the five main crochet stitches. Each row begins with 20 chain stitches + (starting stitch height -1) number of chain stitches to create a neat edge. For example, if the pattern begins with a double crochet, the crocheter would begin with 20 + 2 = 22 chain stitches before creating the double crochet anchored in the 20th chain stitch.

compiler had additional challenges due to its 3D domain and a real threat of failure that could damage the machine. These abstractions allowed the feeling of agency of familiar physical crafts to transfer to design software more easily. The pattern design of blackwork embroidery is immediately recognizable and actuateable due to embroidery machines. The knitting compiler's abstraction enables knitters to more easily visualize their designs and use familiar knitting terminology. Familiarity and confidence increases due to improved presented affordances, which leads to higher agency and ludic engagement. However, our lack of priority in making the presented affordances clear to users via a polished user interface reduced the sense of agency of some users (Figure 3.44).

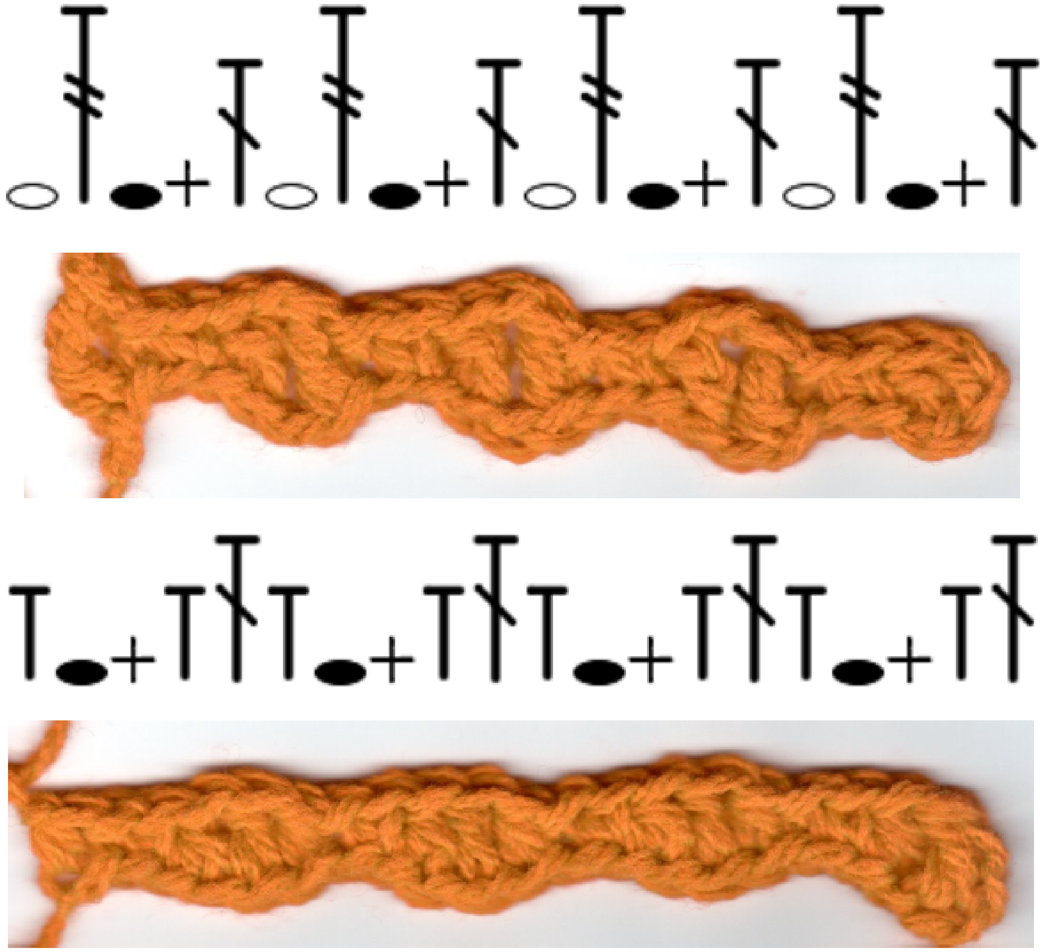


Figure 3.42: A sample semi-random arrangements of stitches using the notation of Figure 3.41

The knitting compiler project shows the editor view for almost every knitted project beside their physical output. For some of the projects, the view somewhat matches the output, and a quick explanation that:

1. the diagram is a timeline of machine operations from top to bottom and
2. grey spaces are areas where no active stitching is happening, but there are stitches on the needles in those positions

should be enough to allow anyone to somewhat read and understand the diagrams.

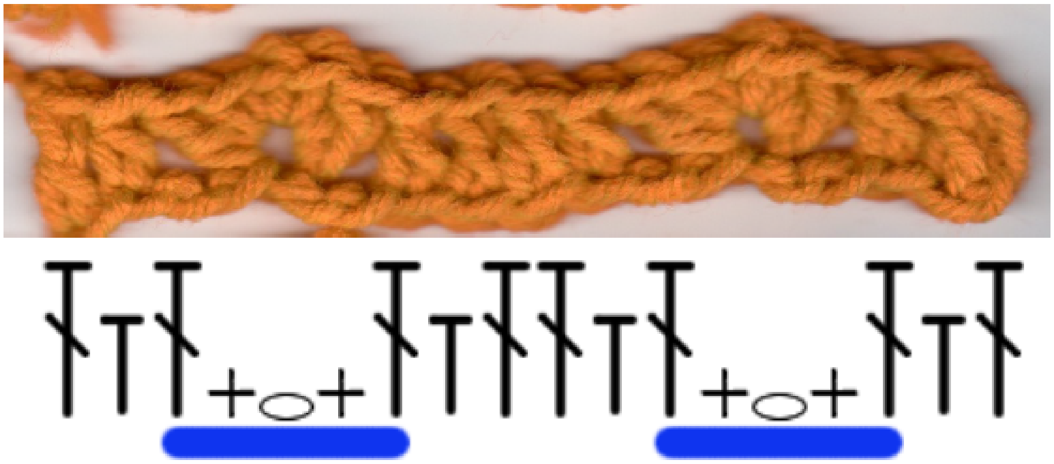


Figure 3.43: A sample arrangement of stitches using the notation of Figure 3.41. The pattern section underlined in thick blue are “clusters:” multiple stitches anchored in one stitch, a common crochet design pattern. To maintain the shape of the row, clustered stitches skip anchor stitches before and after to equal the total stitch count. For example, a cluster using 5 stitches would skip 2 anchor stitches, stitch the entire cluster (5 stitches in this case) in anchor stitch 3, and then skip 2 anchor stitches before continuing. The clusters “fan out” into the empty space left by skipping stitches.

A close-up example is seen in Figure 3.45:

However, making use of some operations, particularly short rows, are much less immediately understandable because their shape is so drastically different compared to the final output. The authoring tool is meant to represent accurate authoring with respect to the machine — representing needles and their individual tasks within planning — not a recognizable object. A close-up example is seen in Figure 3.46:

You may notice that the general conceit of KnitPaint in Figure 3.47 is the same as our knitting compiler: a planning interface, top to bottom, of what commands are executed on what needles. However, these operations in KnitPaint are the lowest-level found abstracted in the knitting compiler; commands such as tuck, split, and drop must all be painstakingly pixel drawn, which often causes further

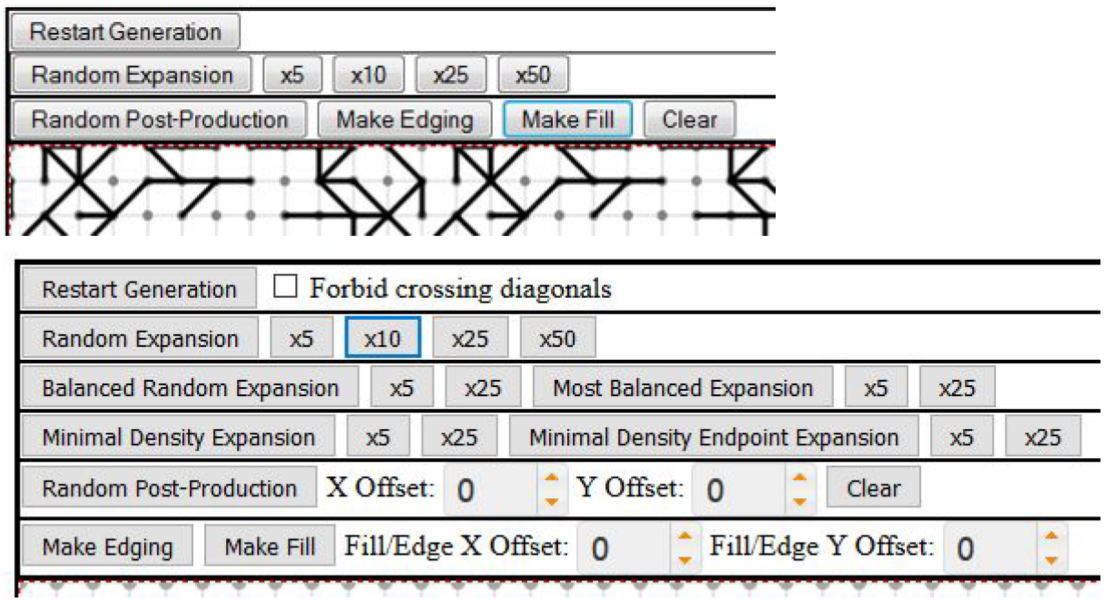


Figure 3.44: Two snapshots of the blackwork embroidery project's interface during development, the top example being an earlier prototype than the bottom. The most obvious next step would be replacing the formal names for what each button corresponds to in the production pipeline in Figure 3.3

warping of the representation of the final product vs the design represented in the software. Regardless of a user's expected affordances of machine knitting, the presented affordances of even a low-level knitting editor suggest much more substance than the low perceived affordances of visual programming via pixel art.

Because the user is given such freedom, there are also little-to-no safety mechanisms involved in this software. In this case, the user's low perceived affordances can cause serious damage. Unrespected yarn tensions could easily bend or break needles on the bed. There is also no concept of 3D in this 2D paint interface. What happens on the front and back beds are depicted in the interface side-by-side. Any side-by-side pairs of stitching of suspiciously similar size, such as the skull design on the right side of Figure 3.47, are actions taken by the front bed on the left and the front bed on the right at approximately the same time.

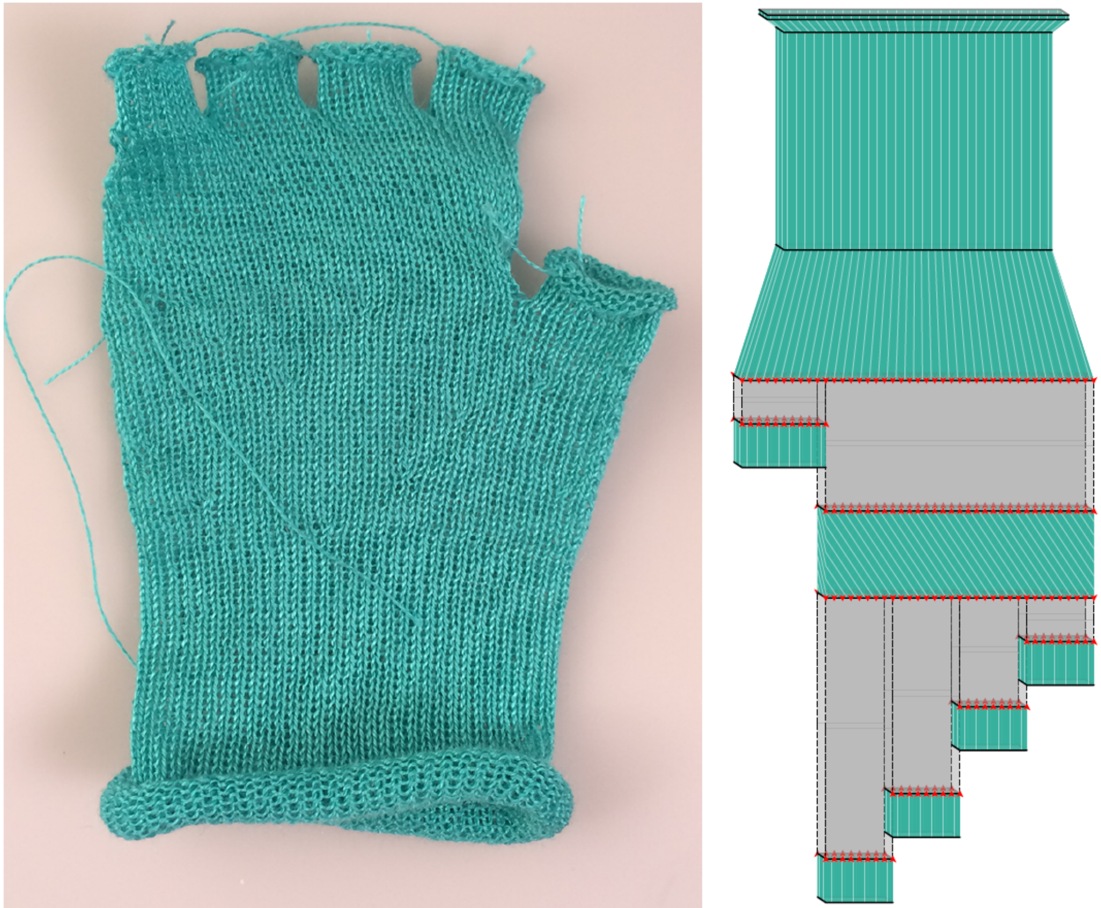


Figure 3.45: The example glove with finger found in Figure 3.33. While the width of the diagram seems exaggerated, that is the property of knit fabric to stretch (and shrink) more horizontally. The fingers are placed where you'd expect, while understanding that the grey is unstitched means that the fingers are not as extended as they may first appear.

3.4.3 Effective Design Software

As a given, software should follow general user experience design principles in their design software, some of which are common-sense, while others are based on principles discussed in sections 2.2 and 3.1.1: enabling common tools such as undo/redo and saving/loading, prioritizing clarity, using familiar tropes and metaphors, narrowing options to relevant choices, and so on [19].Knowing your domain and the features you have allowed and cut from your design space should

Figure 3.46: A knit sock, with short rows only on the toe and heel. Because short rows do not increase the circumference of the whole object, the view of the needles on the bed does not widen or bend in any way. While this view is more intuitive than the standard Shima Seiki interface (Figure P1AG), it is still assumes more technical knowledge than most users would be comfortable with.

help drastically reduce the number of options and thus the cognitive load on your users. Less options, while still keeping those options featureful, keeps the tool simple and friendly to amateur users. For newly created design software, all of your users will be novice to at least your software, if not to the crafting domain at large. Supporting novice users should then be a priority, which includes buffering them against failure wherever possible.

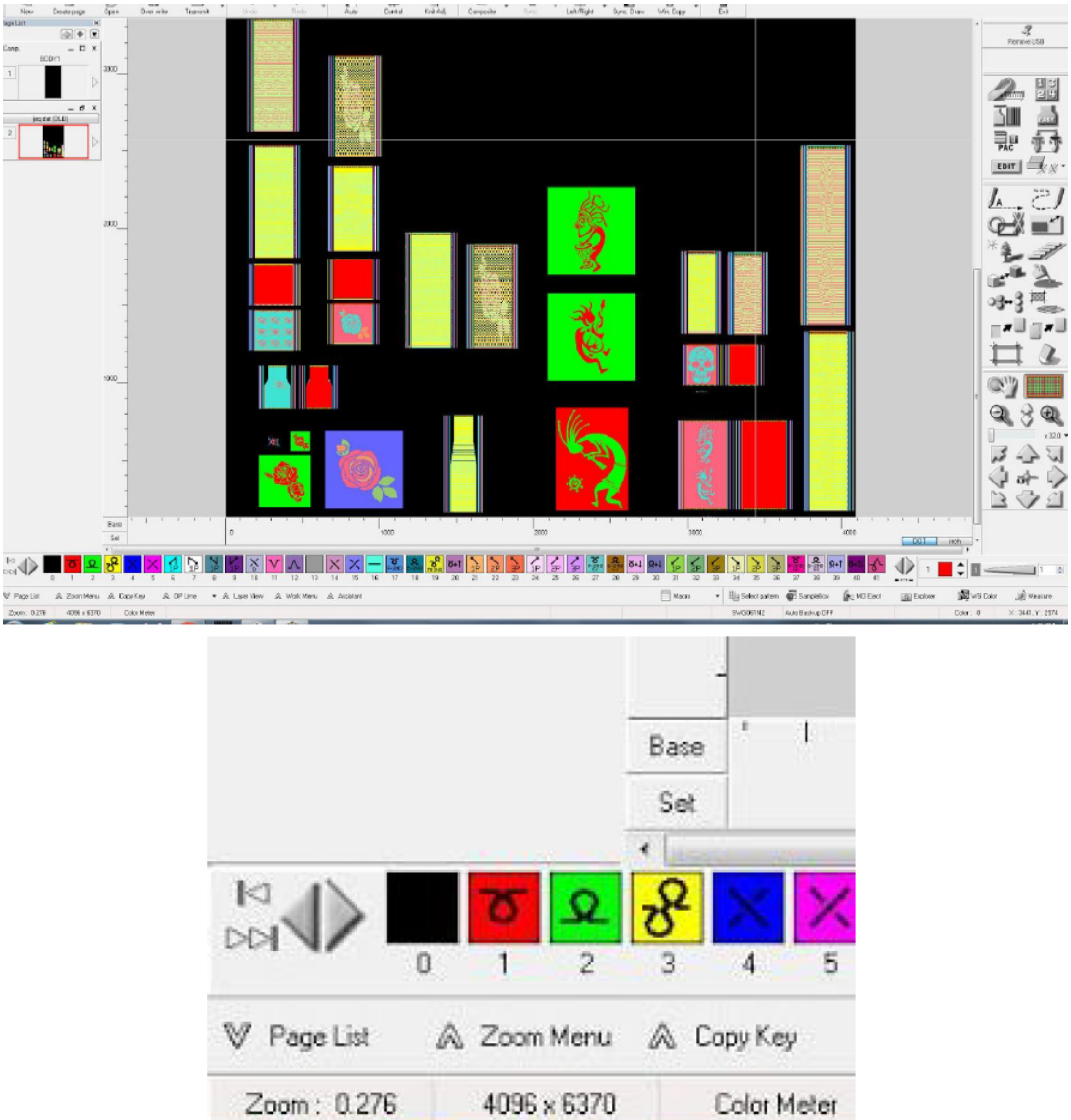


Figure 3.47: This is proprietary software that comes with a Shima Seiki knitting machine, SDS-ONE KnitPaint. The top image is an overview of an open project, while the bottom image is a close-up of the colored pixel legend that is in the lower-left of the top image. Each colored pixel represents a different low-level machine command directly for the knitting machine. For example, the red pixel, labeled “1” is a knit motion, while the green pixel, labeled “2” is a purl motion. Their placement on the grid indicates which of the dozens of hooks on the knitting machine performs the motion. There are many designs on this screenshot [61] that are not intended to be knit all at the same time.

We have previously seen an atrocious design software example in KnitPaint in the previous section. The domains of different crafters are very disparate, but let us examine a case study of a different domain and a more successful textile design software that takes full advantage of the 2D aspect of quilt pattern design.

Electric Quilt

The first version of Electric Quilt (EQ), designed by Dean Neumann, debuted in 1991 as part of the "Great American Quilt" program (WBGU-TV), Figure 3.48 [39]. In the segment, the same motivating design principles for this chapter — rapid iterations, outputting executable patterns — are expressed:

> ...in seconds you can see what your finished quilt design is going to look like... so rather than making all the blocks and then crawling all over your living room floor twisting and turning them, you can let your fingers do the twisting and turning.
>
> ...You can see those ideas almost as fast as you can think them up.
>
> ...The computer will even split up the patterns for you to any size that you want."

Like the blackwork embroidery design approach in this chapter, there are common quilt block configurations that can be visualized via repetition, reflection, and rotation, and they look drastically different based on the composite block(s). Cycling through these design possibilities is the joy of using these programs. The exploration of the design space offers high ludic engagement, as the base features of the tool support safety, freedom, and a large possibility space to explore. It is useful to note that this design software was marketed to people who may not have ever even used a computer. A handful of key commands are shown on every screen, and each screen has an explicit purpose and very few options.

All parts of the interfaces come with defaults and randomized options, so users never have to define any or all parts of their quilt blocks, and they can always

Figure 3.48: A snapshot of Electric Quilt 1 from [39]. The host, Penny, spends much of the segment pseudo-randomizing the layout of the log cabin block, or cycling through a randomizer of existing blocks, colors, borders, and layouts.

save their favorites. EQ1 was the first software in its series, where every user was a novice to the interface. The users that would have been most interested in using the tool (those that watch a quilting program) would likely have had physical quilt experience, a feeling of agency at least, that would support user expected affordances and their ability to perceive the presented affordances in this tool.

Obviously the design has matured since 1991. Electric Quilt 7 (EQ7) (Figure P1XA), the second-most-recent design software, is presented here as a contrast to the following Electric Quilt 8. Its overall design looks much more like a standard paint program, with layers, bars of buttons at the sides and top, and a wide range of drop-down menus at the top. Sub-menus are pop-up dialog boxes full of options:

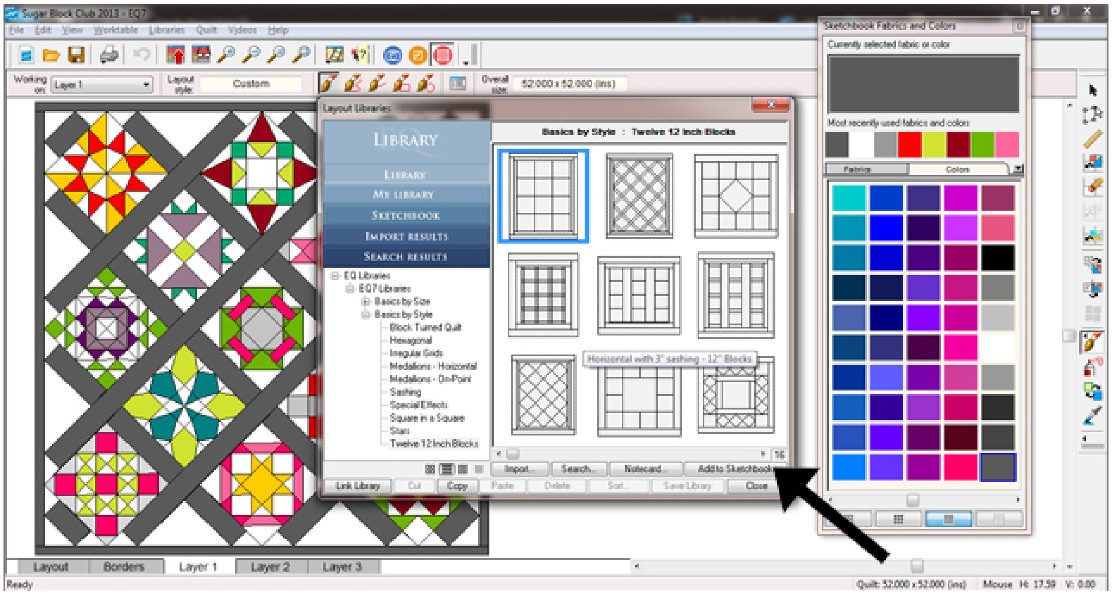


Figure 3.49: Electric Quilt 7: Layers of menu trees and endless options, which is customary in complex textile editing software. Machine embroidery editors look very similar. [74]

There as been a great opportunity for novice users to become experts between between Electric Quilt 1 and 7. These expert users grew accustomed to all the available tools, which only got more featureful and complex over time. However, the most recent iteration, Electric Quilt 8 (EQ8), has been lauded for its particular focus on its iterated user interface. UK Quilter's Guild's blogger, Chris Franses, points out that the interface has been changed so drastically that users of previous versions often do not like the changes, themselves included [68]. However, another blogger explicitly calls out the similarities to EQ1 in the EQ8 design philosophy, and that the redesign is much friendlier to new users [163]. The buttons are bigger and labeled clearly without mouse over or additional tooltips. The tools are sectioned into clear tabs that represent stages in design. Even the opening of the software simply begins by asking the user the explicit question: "What would you like to do?" and offers stages of the craft process (Figure 3.50).

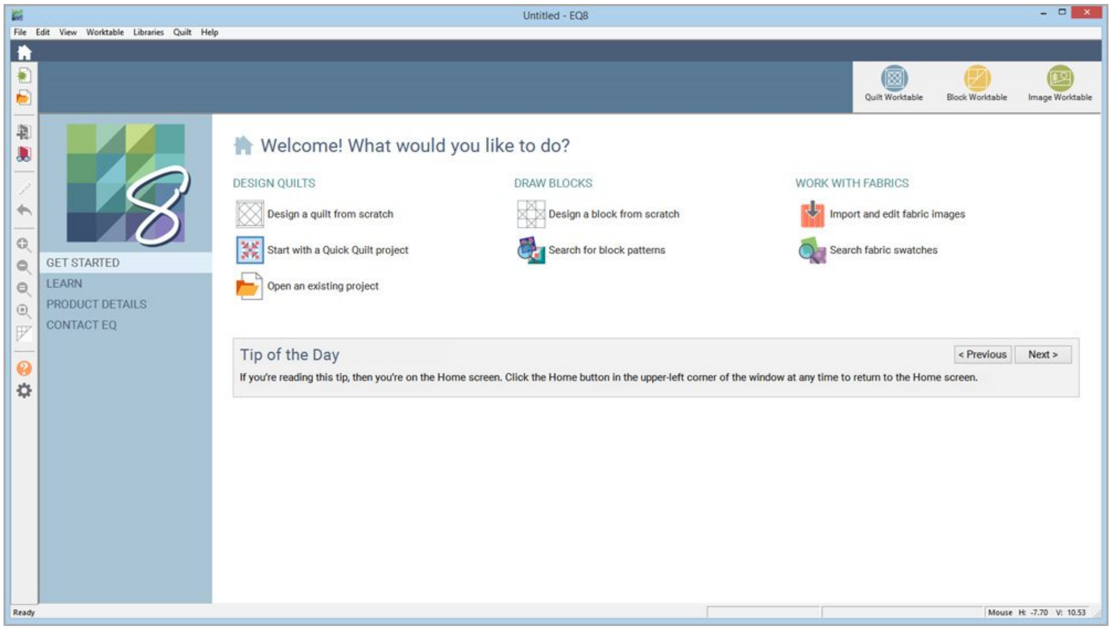


Figure 3.50: EQ8 greeting. Understanding a user's goals (and thus their needs) is the first step to offering targeted support, which EQ8 aims to explicitly accomplish from first interaction with the user.

In general, both digital and physical quilt designers understand that there are defined phases to the quilt design process: you draw when you're designing your blocks (even if there is only one block), you play with color within blocks, you play with color and positioning amongst blocks and borders. You may swap between these stages fluidly, but each involves high level tasks and goals. Newer versions of EQ accomodate more quilts styles and custom fabrics, and incorporate arguably the most accurate yardage calculator of quilt design software. However, the overall design of EQ8 has also gone back to acknowledging that usability for novice users will help keep the quilting community using their software thriving.

This chapter explored how different domains and design decisions in making software-based design tools affects the agency and ludic engagement of their users. The properties of successful creativity support tools discussed in section 3.1.1 are demonstrated here as elements that support agency and engagement. Ease of use and familiarity to the physical crafting domain enable users to feel comfortable and confident in exploring the design space, which encourages ludic engagement and prompts expected affordances. High-level abstractions of low-level machine or other design operations allow the design tool's presented and perceived affordances to more closely match the user's expected affordances, which increases the user's sense of agency. The design tools represent physical crafts, which allows the physical and digital agency of the user to support each other. The next chapter more deeply explores this relationship between digital and physical agency and how it can be shared with machines during the craft manufacturing process.

Chapter 4

Perspective 2: Textile Craft Manufacture

Humans and machines can read and execute patterns in most textile domains (we discussed how crochet is an exception to machine manufacture in section 3.4.1). Perspective 1 focused on generating and designing patterns, while the two projects in that section elaborated on two examples: blackwork embroidery for hand and machine execution, and knitting instructions for industrial knitting machines. The high-level abstractions made in the design tools aligned with physical crafting domains and provided safety and confidence in the user's exploration of the design space. Overall, when a domain is chosen and a tool is executed well, the design tool supports user agency and ludic engagement, This chapter examines how both humans and machines interpret, improvise, and make physical artifacts, as well as how digital and physical agency support one another.

4.1 Textile Craft Manufacture Research Review

4.1.1 Machine Design, Human Manufacture

A human designing patterns for other humans, or a human using the authoring tools described in Perspective 1, are the commonplace defaults for textile crafts. In reference to section 2.2, many participants in creative computing research hold biases against machines and their ability to participate as a full-fledged collaborator in a creative process. However, fully or mostly giving up authorial control of a pattern to a computer or algorithm is a means of forcefully demarcating the crafting process. The importance of the role of computation in the following creative projects and their processes is undeniable.

Hoopla and Foundry

Following the generative systems and the generated designs from Perspective 1, the obvious next step would be the human manufacture of these generated patterns. Taking the next step to bring a digital design into the physical world shows a high confidence in both the design and the user's capability to create it, which suggests high digital and physical agency. Gillian Smith created two such generative systems: Hoopla, which generates cross-stitch embroidery samplers[1], and Foundry, which generates quilt patterns inspired by foundational paper piecing techniques[2] [180]. Both of these generative systems are designed for human production, as directly stated by Smith.

[1]Embroidery samplers hold a long tradition in embroidery as a means for crafters to explore different stitches and motifs. Cross-stitch embroidery samplers are embroidery samplers made in the cross-stitch style.

[2]Foundational paper piecing is a quilting technique inspired by the hex-based hand-quilted style (that uses paper templates to hold the hexes in place). Foundational paper piecing similarly uses paper patterns that support the fabric while the user sews through it before later tearing the paper away.

Human manufacture requires human-legible patterns. Luckily, evenweave embroidery techniques, such as cross stitch and blackwork embroidery from the last chapter, are easily legible when presented on a grid (more detail in section 4.2.1 below), and a simple quilting paper piece template with only a few subdivisions is also generally easy to understand, with numbers representing a sewing order of fabric pieces. That is to say, it is not easy to make the framework or generate these patterns, but not much detail needs to be added to the graphical confirmation of the generative process for a human to interpret it. However, interpreting the pattern and spending the laborious hours to manufacture it are two entirely separate skills and tasks.

Smith summarizes the purpose of the generative systems in [180]:

> The blending of digital and physical spaces, the tension between machine and human authorship, and the juxtaposition of stereotypically masculine computing with highly feminine textile crafts, leads to the opportunity for new kinds of tools, experiences, and artworks.

The work presented in this chapter and dissertation as a whole shares this same philosophy.

SkyKnit

Alternatively named "Project Hilarious Disaster" by its author, Janelle Shane, SkyKnit is the result of a neural network trained on 5228 example patterns [117, 179]. The neural net is able to reconstruct the common structures and terminology, like *rows, p, k, sl*, and even meta structures such as **repeat* notation. However, consistency between the number of stitches in a row or any high-level logic of structure does not exist.

The system outputs "were just on the edge of knittability" [117]. One participant playtesting these output patterns described a common predicament, "'If you are knitting along and have 30 stitches in the row and the next row only gives you instructions for 25 stitches, you have to improvise what to do with your remaining five stitches"' [117]. Many knitters that have attempted to follow patterns have no doubt come upon minor mistakes or inconsistencies. However, the lack of intent behind the pattern — is it a pattern for a shawl, a sleeve, a scarf? — significantly hampered user's overall confidence in their interpretation of the pattern.

However, compared to the patterns made by Hoopla, Foundry, or the blackwork embroidery project (see below in section 4.2.1), participants knew they were knitting extremely flawed patterns. For this particular project, the participants' expectations were not high or very specific; one user, agadbois, described the output as potentially a "godforsaken mangled bit of fabric" [117]. Many users encouraged and helped Shane gather data for the training, and they clearly communicated their excitement on their Ravelry [152] group forum, LSG (lazy, stupid, godless). Simply participating in this technological experiment was an unusual and fun activity for the dozens of people who interpreted, knit, and shared their results from the generated patterns.

The next neural net textile craft experiment on Ravelry is HAT3000, a crochet hat pattern generator. As of this writing, Ravelry is starting to see its first outputs in fan projects on the LSG knitting group forum. Note the far more reduced domain space, crochet *hats* only, which follows from section 3.4.1 in Perspective 1.

4.1.2 Fabrication Technologies

The advent of programmable manufacturing technologies has the potential to enable a variety of recreational and product-focused activities. However, most work with these technologies has focused on manufacturing needs. Just as user interface design has transitioned to include values such as enjoyment, aesthetics, and strategy, so too is there room for manufacturing technologies to explore these values, thus enabling new user experiences. New user experiences encourages ludic engagement through curiosity, as well as expanding the user's knowledge of agency.

3D Printing of Soft Interactive Objects

Needle felting is a unique craft that takes advantage of the physical properties of wool to entangle fibres together using a felting needle (Figure 4.1). The general process of felting involves stabbing through layers of wool fibres to intentionally and deeply entangle them, actively 'fusing' them together into one piece of fabric. Alternate methods of felting exist, but using needles is the primary way of both industrial felting and hobby 3D felting. Industrial felting involves a bed of needles, similar to the knitting machines, that punch layers of fibres hundreds of times into one flat sheet (such as those seen at [67]).

However, Scott Hudson performed a materials exploration on 3D printing using textiles, specifically using wool and wool felting methods [86]. Hudson desired to find a means of soft and flexible 3D printing, which, given the materials, would be fuzzy and inexact compared to current means of 3D printing. The machine Hudson built primarily uses a felting needle and a lockable yarn feed to needle felt the wool yarn onto a starter felt base. Hudson integrated various designs for pockets for electronics with explicit support for e-textiles in mind (see chapter 5). The design of the printer is done using laser cut and 3D printed parts, which are common fabrication machines in current fabrication labs. This wool-based textile 3D printer demonstrates a new way to integrate machines into the crafting process to make customized designs. However, this generally removes the human from the

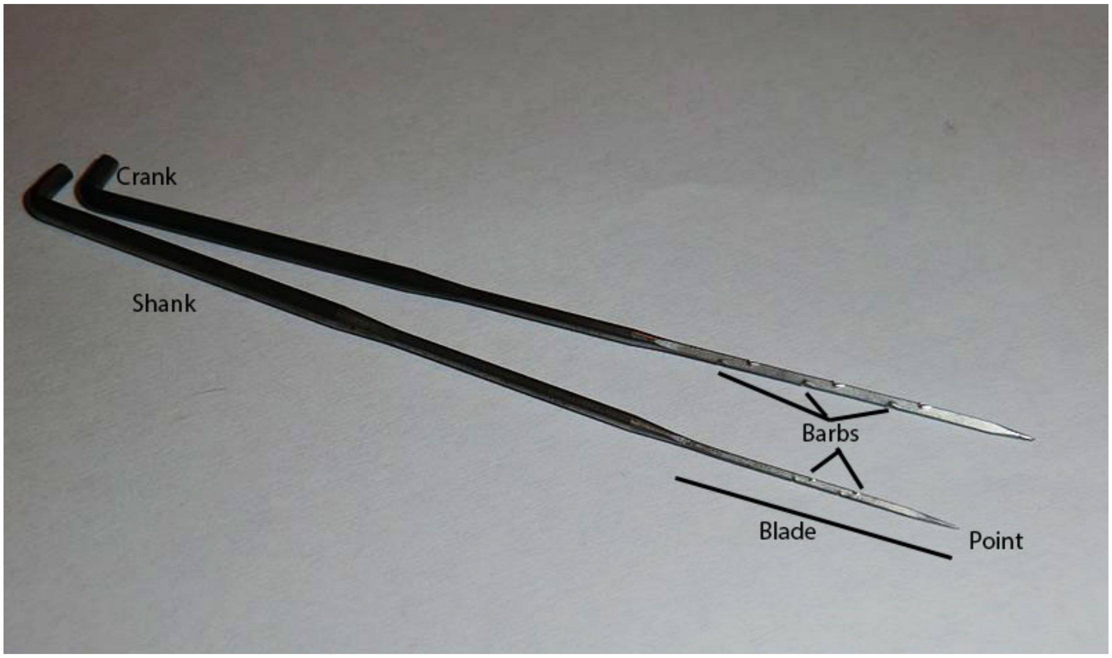


Figure 4.1: The anatomy of a felting needle [142]. The point allows the blade to move alongside loose wool fibres, where the barbs catch the fibres. The fibres are then carried deeper along with the point until being deposited when the blade moves backwards. The 90 degree bend at the top of the needle (the crank) indicates that this needle can be used on industrial machines.

needle felting process.

Spyn

Daniela K. Rosner and Kimiko Ryokai published two qualitative studies of their augmented knitting system, Spyn [159, 160]. In the first study, seven knitters used Spyn, “a system that associates digital messages with physical locations on knit fabric” [159]. The participants used a mobile kit that included a rotary encoder, as well as media recording and GPS[3] equipment. As the participant knitted, the yarn was drawn through the rotary encoder and printed with invisible infrared ink that would read as long and short stripes. When saving media, the participant held their knitting up to the Spyn system, which read and stored these

[3]Global positioning system.

bar-code-like markers and left direct evidence of their increased engagement with the craft. Users' media (images, sound, or video) was then recorded, associated with their location and the time, and was able to be recalled at a later time. The Spyn kit was given to the study's participants and was used without the presence of a researcher.

Some of the participants' recipients for their knitted items were babies or had yet to even be born. The participants were able to record messages they knew wouldn't be seen or heard for many years, and the recording of these messages was very emotional. The participants also reported increased pressure, that the artifact they were making held additional emotional baggage, where one participant "referred to her knit as 'emotional blackmail"' [159]. Other participants that knew they or a friend would be receiving the item left playful messages. Even if the user did not feel the need to get engaged, they showed the most ludic engagement under these feelings of lessened pressure.

In Rosner and Ryokai's second field study of 12 participants, the media capture devices and custom computer interface was replaced by mobile phone software. Through lessons learned from the past user study, the infrared ink was replaced by computer vision within the software, and many additional editing features were added such as the capability to knit multiple projects at once and add multiple media entries to one pin. The second study is much more stream-lined in terms of its tech, more thorough in its participants surveys, and extended to interviewing recipients of the items (as it was much easier for them to retrieve the data given the mobile phone application):

> Using Spyn, creators left behind digital and physical traces that heightened recipients' appreciation for the gift and enabled a diverse set of meanings to emerge. Digital engagements with Spyn became a means for unraveling the value of the gift: recipients used digital information associated with the physical objects to interpret the story behind the

objects and their creators [160].

The Spyn project represents a long and in-depth examination of the crafting process, with particular focus on the crafter's feelings, intentions, and thought processes, especially in relation to the recipient of the craft. The content of the recorded media changed drastically based on the relationship between the crafter and the recipient, which clearly expresses the breadth of personal relationships with crafts, even in a study with so few participants. The deep relationships with the crafted items during and after creation leaves direct evidence to the user's ludic engagement.

4.1.3 Games with Fabrication

Game design's affinity for creative engagement and interaction make it a promising domain for investigating innovation in the use of manufacturing technologies. Alternative input controllers, like the MaKey MaKey (see section 5.1.1), allow physical objects to be used for game input; manufacturing technologies can make game *output* physical in a dynamic and permanent way, offering a new kind of game experience that cannot exist without these technologies. However, supporting playful experiences requires adapting machines that typically take a single piece of input and produce a single artifact into ones that can support interactive control and progressively refine an artifact.

Realtime, situated interaction methods are less common for fabrication machines, which typically take as input static 2D or 3D models. An exception is Mueller et al.'s Constructable, which permits interactive control over a laser cutter through the evocative use of a laser pointer [133]. Textile fabrication machines are even more uncommon in non-commercial and research contexts, given the lack of technology to support crafters outside of industrial machines, as well as the historical stigma of textiles being a feminine domain, which does not often overlap with the male majority in hobby electronics and game design spaces. The following games integrate the human's crafting process and games in an innovative and playful manner.

Loominary

One project that closely aligns with our playful interactions with textile manufacture in this dissertation and chapter is *Loominary* by Sullivan et. al [191]. Players play a choose-your-own-adventure-style game made using Twine, where their choices in the narrative are assigned specific colors. In order to enact that choice, the player must weave a row of yarn of the choice's corresponding color on *Loominary*'s controller: a tabletop rigid-heddle[4] loom. "In this way, the player's choices are literally woven into a physical record of their play session" [191].

While the narrative is presented digitally, and RFID[5] tags attached to the shuttles are the computational means of detecting a choice, it is the human players that must follow the pattern made by their choices and make the craft itself. *Loominary*'s struggles with aesthetic presentation, demo feasibility, and the physical properties of their materials match those we had in designing and making *Threadsteading* (section **??**).

BeadED Adventures

Research continued in craft-based Twine story interfaces with a project explicitly aimed at middle school girls to keep them engaged in STEM education [190].

[4]The rigid heddle is a hard horizontal piece, usually a bar or slat, with holes for each weft (vertical thread). Different wefts are threaded through the holes in the heddle before weaving, and the user lifts and drops the heddle to highlight all those wefts at once. Different heddles can be used for different parts of a pattern, and the use of the heddle drastically increases weaving speed.

[5]Radio-Frequency IDentification.

Instead of using a loom, players create a bracelet strung with pony beads, whose colors indicate choices in the same way as *Loominary*, but with enough length that an entire bracelet is created. A special silver bead marks a particular meaningful milestone in the story, such as reaching the ending. Conductive material at the bottom of the jars of beads is used to detect when a player picks up a jar to extract a bead for their choice. Modified lids on the jars ensure users quickly and easily get only one bead for each choice.

Yarn Quest 2017

Tania Richter on Ravelry [152] set up two Mystery Knit-Alongs (MKAL) to the theme of an role-playing game adventure through the world of Yarnia: a scarf-based campaign of 8 clues [155], and a blanket-based campaign of 20 clues [156]. Before starting one of these, players build a character with stats, as well as personalize a feline companion. Over the span of weeks and months, clues and bits of story are released, where the player rolls dice and faces challenges to their personally designed character. Based on the results of those rolls, they recieve a colorwork chart to knit until the next clue is released. “There are several dozen charts, so everyone’s scarf will be unique!” A Kickstarter[6] to fund the continuation of this project, *Welcome to Yarnia*, succeeded in 2018 [157].

4.2 Design Software as Manufacturers

Both projects from Perspective 1 had a primary goal of being manufactured by their respective machines. While the design software supported ludic exploration of the design spaces, the software from Perspective 1 generated machine patterns. The human may have agency in operating these machines separately from the

[6] A public fundraising and crowdsourcing platform.

design software, but I have yet to explore how the designs foster physical crafting agency within a human interpreter of the design software's patterns. The following section briefly reviews their machine and pattern capabilities and evaluates how well their domain-specific languages translate to human interpretation.

4.2.1 Blackwork Embroidery Pattern Manufacture

The blackwork embroidery pattern generator operated on an 8-way linked list data structure that represented a graph, where edges between the graphs were stitches that would be sewn. The generation algorithm took input rule specifications by the user, ranked and selected expansion rules, and made additions to the existing design. The user then played with rotation, reflection, spacing, and repetition options to create a final design. After this generation pipeline, illustrated in section 3.2, the result is a series of connected subgraphs, where their connected edges are stitches that need to be sewn (Figure 4.2). The visualization is the same, where edges on the graph represent "on" black lines on the grid. I explored various direct properties of graphs, such as neighbors and cliques, but they were not at a large enough scale to impact the design. Higher levels of graph structure analysis would be a direction for further expansion of the grammar.

For the Machine

Applying a depth first search algorithm to the graphs enables us to find a series of adjacent points to sew. However, converting it to a file legible to an embroidery machine can be complicated. Embroidery machines come from a wide range of manufacturers and use over two dozen different file formats. The DST format was chosen for its simplicity and ubiquity: it does not hold colors, vectors, nor images, only focuses on the plotting data for the machine, and is readable by

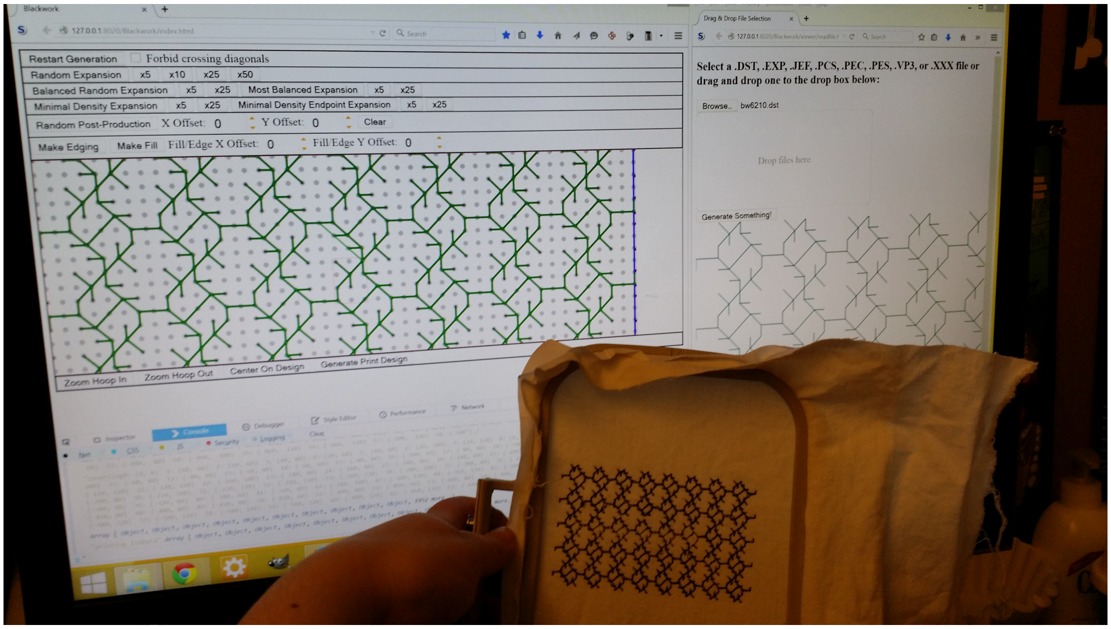


Figure 4.2: The blackwork design software (left) has a green overlay, generated by the depth-first search algorithm, that shows the path of the needle. The separate verification software (right) shows the file saved and loaded correctly. The hoop in the foreground is the design as it was sewn.

every embroidery machine I have encountered.

Thanks to the EmbroiderModder project [2] and a public mapping of its raw data files [42], it was fairly simple to design and confirm through stitching whether my DST file format generator worked as expected. Even though I could theoretically generate any sewing design by having this file format under my control, I kept the domain very restrained due to the authorial challenges described in Perspective 1, section 3.4.1. Figure 4.3 shows designs made by other users during an informal demo setting. The users were able to determine the available grammar rule expansion options had different stylistic outputs, but were not able to easily deduce how or why.

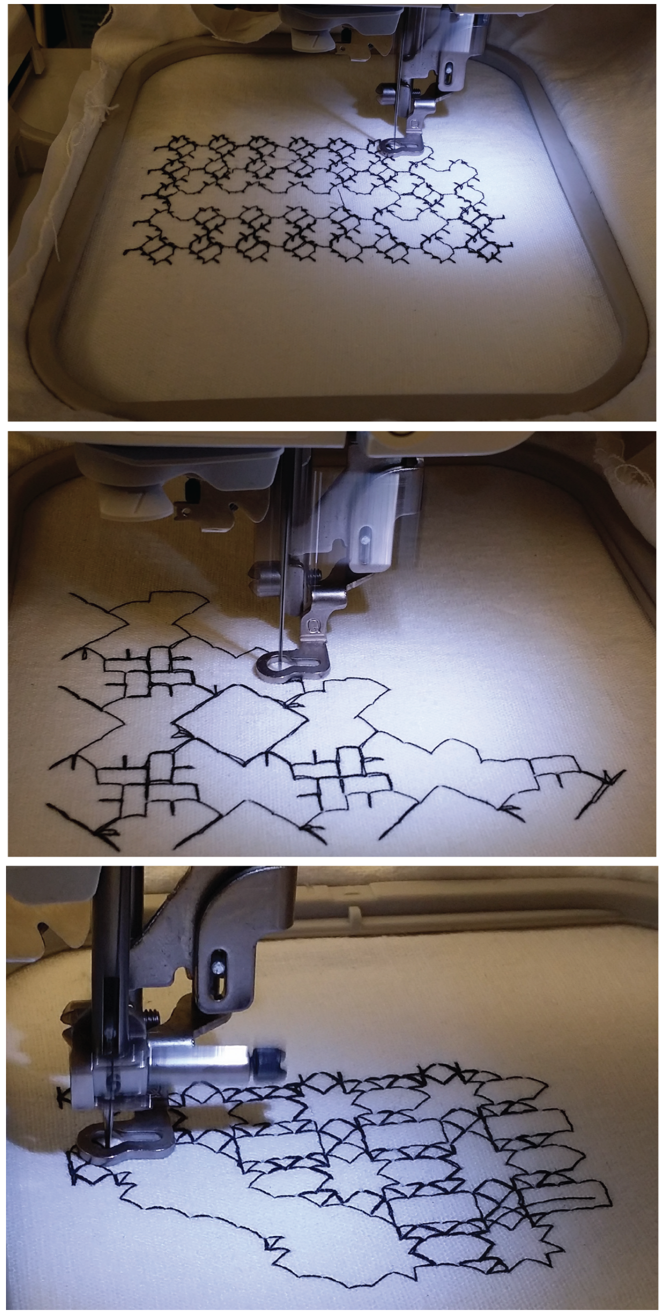


Figure 4.3: Designs made by users in an informal demo exhibition.

For the Human

Embroidery patterns for human manufacture are easy to interpret if the pattern output is an image: grid-based designs for even weave fabrics easily fit into

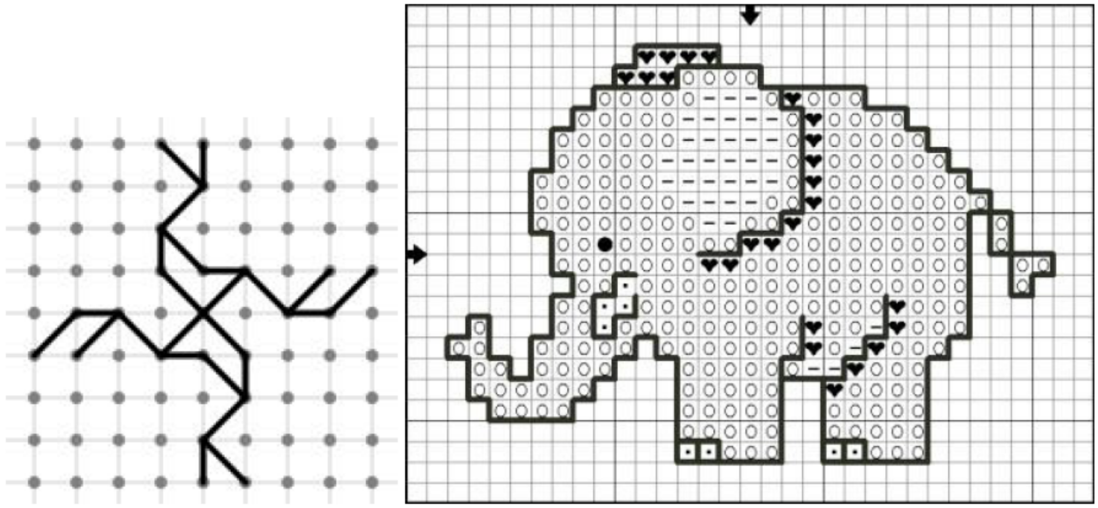


Figure 4.4: On the left is an image from the blackwork embroidery pattern generator. On the right is a cross stitch pattern, where different symbols are cross stitches in different colors, and the bolded lines are outline stitches that work exactly like the blackwork patterns. The similarity to the cross-stitch style of embroidery is one reason why forbidding crossing diagonals is an option in the blackwork generation parameters (section 3.2.2).

pixel-like representations (Figure 4.4), and general embroidery instructions are commonly given as simple black and white line art (Figure 4.6 and 4.7). Grid-based designs presented in common domain notation not only is a sign of a well-authored pattern (see section 4.4.2 below), but increases a crafter's confidence in interpreting the pattern using their knowledge of agency. Hoopla's output does not present itself on a graph, which makes it more difficult for a human to follow exactly (Figure 4.5 vs. 4.4). However, Smith indicates that the patterns are not necessarily made for exact copying, and that improvisation is a natural part of interpreting crafting patterns [180]. Improvisation is a common and playful activity among crafters following patterns: a compromise between designing a whole pattern from scratch and making an exact copy of someone else's designs. Custom interpretation not only leads to ludic engagement with the pattern design process, but tests the physical agency of the now-designer in relation to their crafting

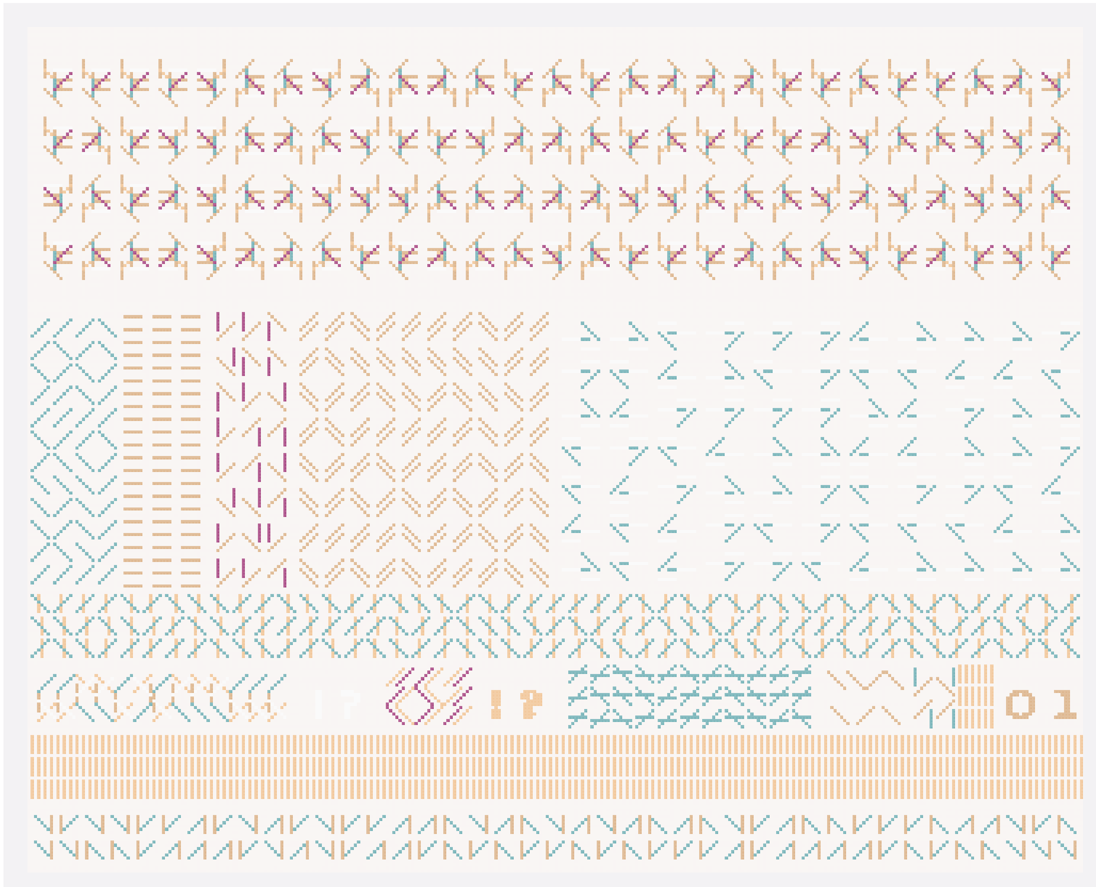


Figure 4.5: An output image design from Hoopla.

domain.

4.2.2 3D Machine Knitting Manufacture

Machines may sometimes mimic the human approach to manufacture, but often they work on their own low-level instructions. For example, instead of two beds of knitting needles that can freely pass stitches across each other, knitting needles hold rows of stitches and generally only operate on the end of a needle where the active thread is (Figure 4.8). However, humans can freely integrate other tools and techniques, offering a broader domain of craftable items.

While much research was done during the knitting compiler project to output

Figure 4.6: A sample of free-form embroidery designs, and iron-on transfer pattern: Vogart Embroidery Transfer Pattern 262: Kitten Honeymoon Motifs Day of the Week Tea Towels [209]. These types of embroidery patterns were very common in the 1950's, where the crafter would iron the tissue paper patterns (bottom left), which were printed with heat-activated ink, onto fabric. It was up to the embroiderer to choose what colors and stitches to use in outlining or filling the design, with simple instructions given with the pattern (bottom right).

Figure 4.7: A sample of free-form embroidery designs: Urban Thread's UT11804 [203] (right) and its hand embroidery version (left). Modern machine embroidery companies sometimes offer hand versions of their patterns in the same form as Figure 4.6, where it's up to the crafter to trace/transfer the design and sew it however they like.

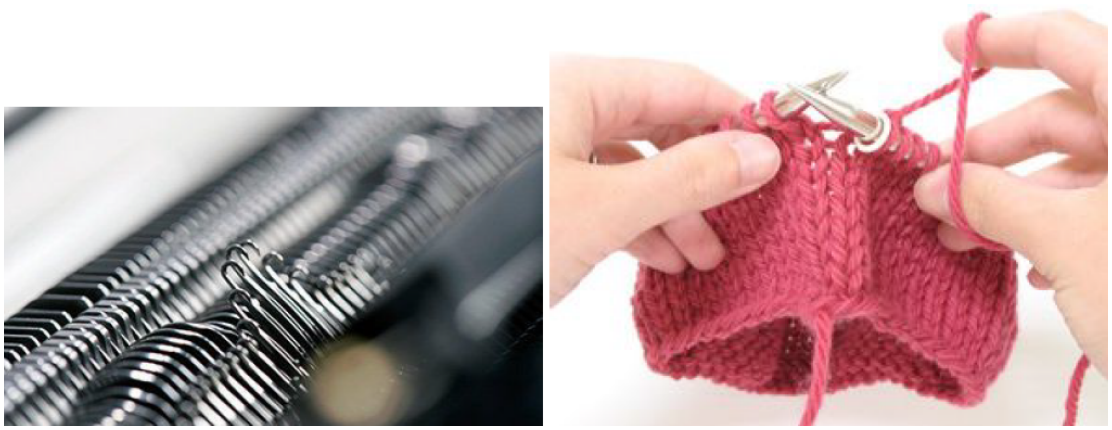


Figure 4.8: On the left is an image of one partial bed of 'needles' extended, while the back bed is in waiting [169]. Both sides are engaged when knitting in the round. On the right is a sample knit in the round on circular needles, showing only one point of contact [199]. On simple examples such as a tube, the differences are minimal, but the whole needle bed management and scheduling is useless to a hand-knitter.

to patterns that could be posted on hosting websites such as Ravelry [152], the process of translating our knitting compiler language to human-legible knitting pattern language was not in scope for that project. As discussed in Perspective 1 section 3.4.1, the agency a designer has for authoring for humans and for machine-level instructions will often be different. The algorithm presented for the knitting compiler works because its domain of operations is much smaller than that of human knitting, and because the operations (while abstracted away) translated directly to machine code. Due to the knitting domain, while authoring with the knitting compiler increases agency in handling the machine, the increased agency does not transfer to human knitting.

The following project attempts to bridge the potential gap in this agency by sharing the manufacturing challenge between human and machine. The human is allowed to take more of an interactive design role while the machine buffers against failure and handles the part of manufacture that would require physical agency in the sewing domain in the user. Reducing the manufacturing burden reduces the barriers to ludic engagement, as well as removing the need for crafting agency. However, the user still needs physical agency: not of manufacturing a craft, but of playing a game, which is much easier to achieve.

4.3 *Threadsteading*

This section (Threadsteading) is adapted from the previously published paper Threadsteading: Playful Interaction for Textile Fabrication Devices *with accompanying authors Lea Albaugh, Chenxi Liu, James McCann, Gillian Smith, and Jennifer Mankoff (for full citation, see [10]). My contributions were primarily toward the quilting machine interface, a minor role in game testing/design, and as a primary showcase demonstrator. I was also responsible for the game's name.*

This project examines the use of textile manufacturing technologies, specifically a computerized long-arm quilting machine and a computer-controllable consumer embroidery machine, for ludic engagement with unfamiliar textile manufacturing machines. We explore manufactured, situated play in the context of two-player quilting and embroidery games. Our work extends prior work in innovative game design by combining situated input on the manufacturing device with physical output produced by the manufacturing device to produce a durable, aesthetically pleasing artifact. This domain is challenging from a game design perspective because interaction is constrained to a device that can only draw a single, continuous line. The game is also constrained by requirements and common aesthetic properties of the final tangible artifact produced during play — in this case, we want a roughly even density of quilting across the game board, with few, if any, lines sewn over more than once, and with the sewing to begin in the center of the fabric.

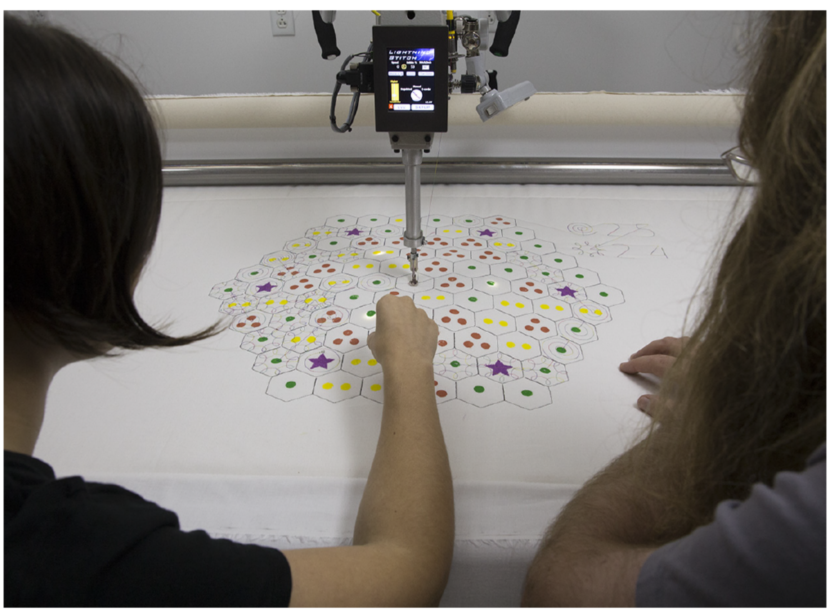


Figure 4.9: Players consider a game of *Threadsteading*. Buttons are attached to the sewing arm under the quilt. In this instance of the game, the game board was sewn and fabric paint stamps for the terrain were added prior to play.

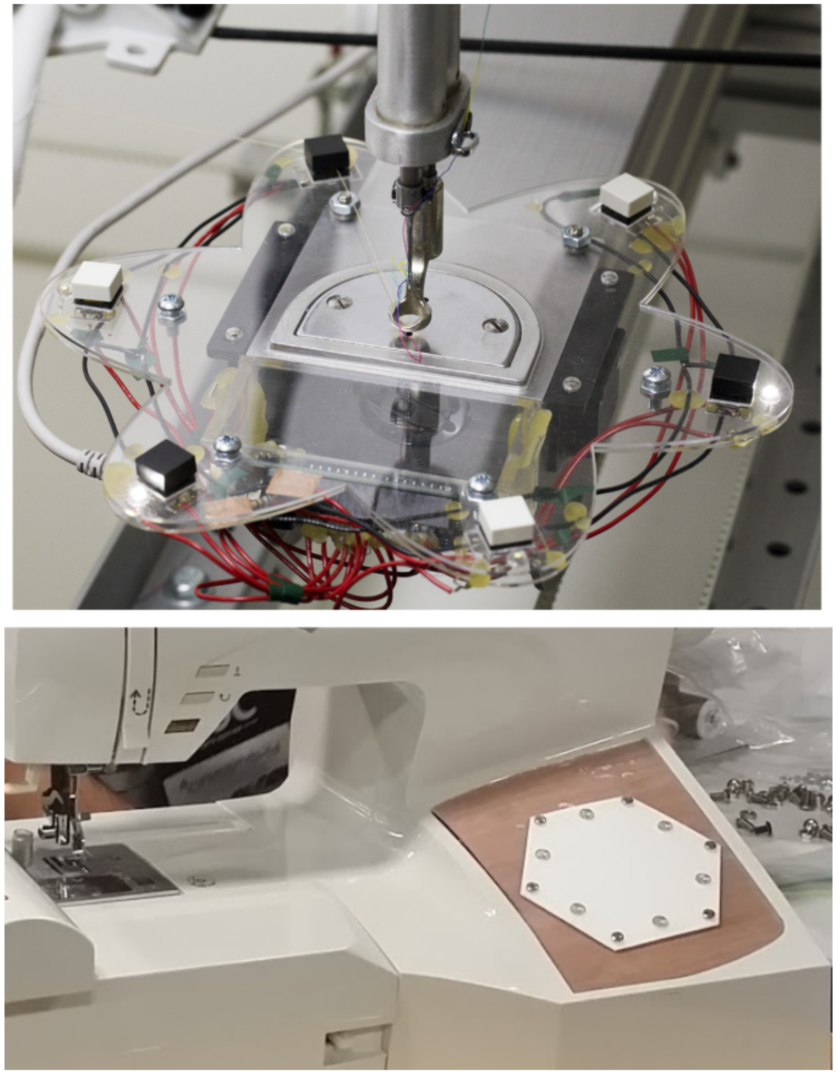


Figure 4.10: The quilting-machine version of our game is controlled with a ring of buttons around the sewing head (top), while the embroidery machine version uses a custom on-machine button panel (bottom). Each turn, the buttons which result in a valid game move light up.

Threadsteading is a two-player territory control game played on computer-controlled quilting or embroidery machine, Figure 4.9. Input to the game is provided through custom on-machine buttons, Figure 4.10, and output is quilted onto a fabric map loaded into the machine, Figure 4.11. See section 4.3.2 below for how the constraints imposed by the use of quilting as a medium and the fabrication processes followed by the machines informed the design of the game.

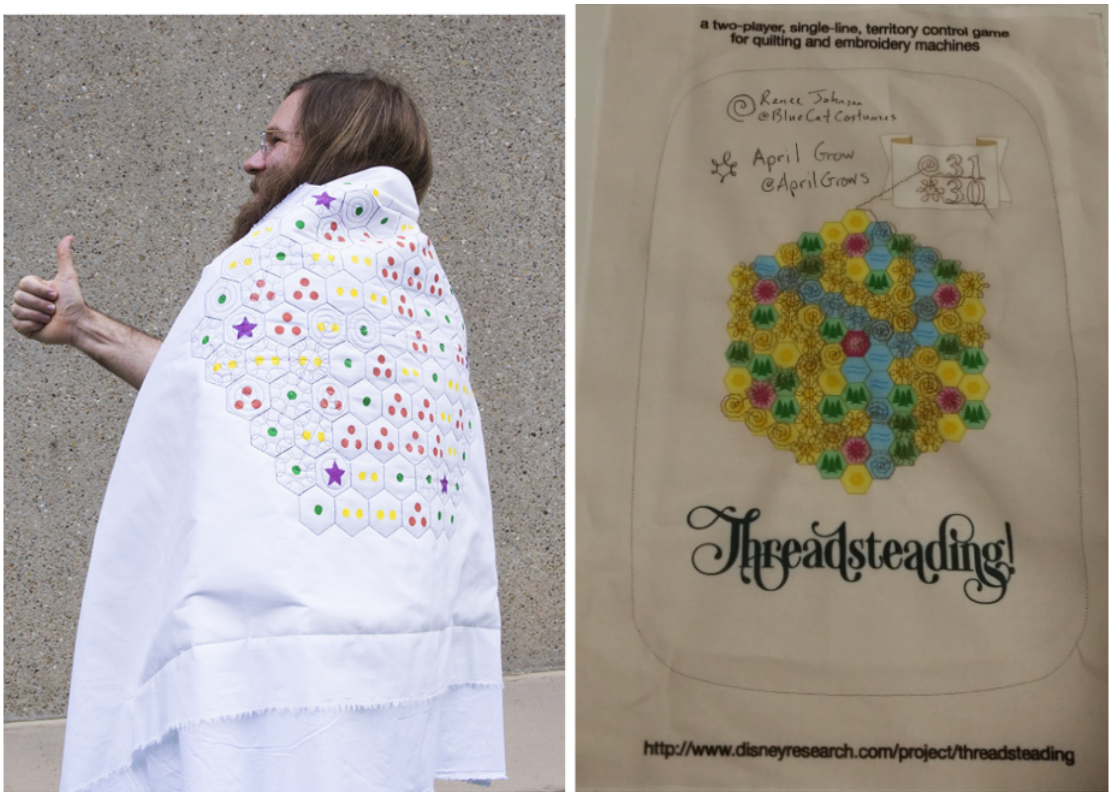


Figure 4.11: The winner of a round of *Threadsteading*, wrapped in their victory quilt (left). The embroidery machine version results in a piece of embroidered fabric that players can take with them (right). The embroidered version required far less monetary investment and time to produce, so we freely gave them away during our exhibitions.

4.3.1 The Game Description

In *Threadsteading*, players take on the roles of rival military officers jointly responsible for scouting territory. Each officer wants to have credit for scouting as much territory as possible, yet they are forced to cooperate to give the appearance of efficient exploration of the space.

The game is played on a hexagonal map with a radius of six tiles. Each hexagon is marked with a terrain difficulty from one to three, corresponding to how many movement points it costs to move into that piece of territory. In addition, six of the difficulty-one tiles are designated "towns," and are worth more points.

The officers take turns deciding which direction their joint corps of scouts should move. The scouts start with four movement points, and march in the selected direction — subtracting the movement cost of each tile they encounter from this total — until they run out, at which point it falls to the other officer to pick a direction to scout in.

Direction of movement is indicated using a hexagonal arrangement of buttons on the gaming machine (Figure 4.10). Army movement is recorded by the machine in the form of a sewn path of player-specific motifs, as shown in Figure 4.12.

Figure 4.12: As a game of *Threadsteading* progresses, board hexes are quilted (and over-quilted) with player's motifs.

Once all six towns are scouted, the machine stitches a scoreboard (Figure 4.13) showing which officer received the most prestige from the scouting expedition. Prestige is assigned to each officer by adding one point for every regular tile and three points for every town that was scouted under that officer's command. Officers don't receive any prestige from tiles that they both scouted because retracing their steps makes them both look bad. The officer with the most prestige is considered the winner. A graphical summary of these rules can be seen in Figure 4.14.

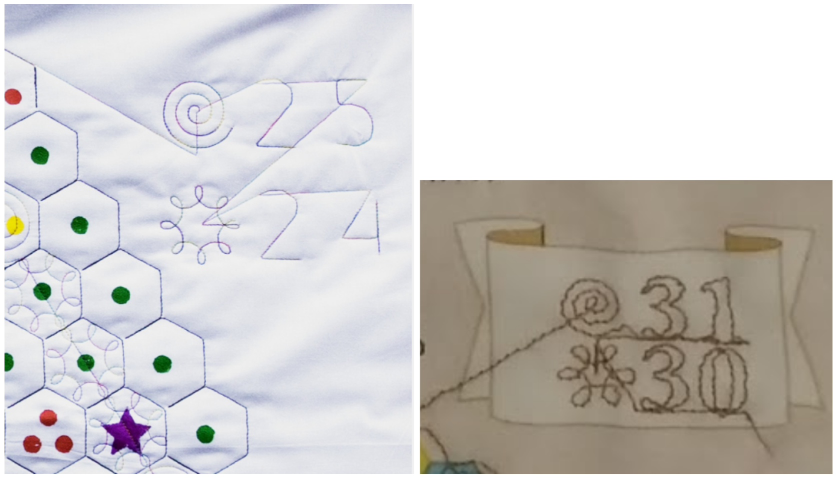


Figure 4.13: When the game concludes, the final score is quilted next to the board (quilted on the left, embroidery on the right).

This introduction presents the game in its final design as it was described and shown at various exhibitions, shown in section 4.3.4 below.

4.3.2 Design Process

Could we make a *game* on a long-arm quilting machine? This was our motivating question as Gillian Smith visited the textile lab at Disney Research Pittsburgh, joining James McCann, Lea Albaugh, Chenxi Liu, Jen Mankoff, and myself during the summer of 2015. We had roughly one week to both design a game for the long-arm quilting machine we had access to in the textile lab, as well as determine how deeply we could control and interact with the machine.

Sewing Constraints for Innovating Game Design

The introduction of constraints into the design process is widely held to produce innovative ideas [27]. There were three main constraints imposed by our use of a quilting machine: the need to sew in a single line, a desire to discourage players from sewing over the same space multiple times, and a goal of producing

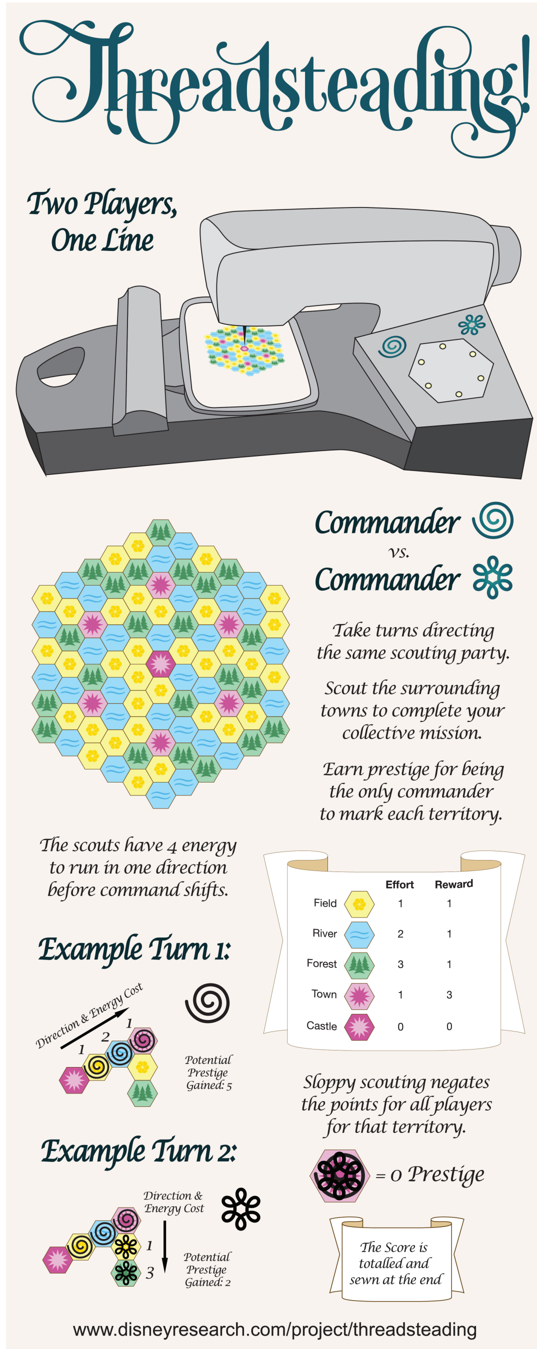


Figure 4.14: A how-to-play poster designed for *Threadsteading* and used at the Sammy Showcase [200] 2016 and IndieCade [89] 2016 (more showcase info below in section 4.3.4). Details such as how much energy each terrain tile costs, as well as two example turns taken on the shown map, are shown in this graphic.

an aesthetically-pleasing, recognizable quilt. Each constraint has resulted in a deliberate game design decision that, combined, resulted in an innovative game tailored to the platform of a quilting or embroidery machine.

Quilting and embroidery machines are best suited to sewing along continuous paths. Moving the needle without sewing either requires cumbersome manual intervention to knot and cut the thread, or makes tangling very likely. Thus, our game is single-path. Each player must start their move where the previous player ended theirs. Both quilting and embroidery machines also favor beginning embroidery in the middle of the fabric and radiating outward in order to maintain even tension and reduce buckling in between stitches. This constraint motivated us to begin the game in the center of the game board.

Repeatedly sewing over the same area can weaken fabric, produce unpleasant visual artifacts, and lead to broken threads or needles. Thus, we must encourage players not to go over ground that has already been covered, which led to the interesting theme of internally competing for territory while externally keeping up appearances. Players are explicitly discouraged from marking the same territory by the rule negating prestige for all players if a game tile is traversed more than once. We also forbid players from immediately moving backwards and negating their opponent's last turn.

Finally, *Threadsteading* adopts the traditional aesthetic of a hexagon-quilt (Figure 4.15). This both determines our board shape, and required us to come up with a goal — exploration — that encouraged adding many motifs to the board.

4.3.3 Implementation

Threadsteading runs on either an Innova 32“ longarm quilting machine or a Singer SE-200 embroidery machine, both of which use proprietary communication

Figure 4.15: A traditional hexagon-quilt (left) [26]. Experiments with playing *Threadsteading* on hex-based fabric or our own pieced quilt resulted in too much distraction from the legibility of the game (right).

interfaces to receive pattern files. Thus, part of developing the game involved reverse engineering these protocols to be able to send paths from our software. We were able to reverse engineer the quilting machine within the original week game jam because both the quilting and embroidery machines are adapted plotting CNC machines. My previous experience in decoding the protocol for embroidery machine files for the blackwork embroidery project (see section 4.2.1) enabled us to systematically discover the protocol that directed stitch direction and distance and work backwards from there.

Both the Innova longarm quilting machine and Singer SE-200 are meant to be constantly attached to a computer running proprietary software. The machines are constantly listening for instructions sent from their respective machines. To control them directly in real-time, we had to hijack these communication pipelines and send our instructions directly to the machines. Without being able to send direct instructions, each turn would be interrupted by compiling and sending sewing instructions for that player's move using the proprietary software. Minimizing the time and additional input between player turn-taking and the machine sewing was a high priority to make the game as seamless as possible. In future exhibition spaces, we did not want to interfere with players every turn, nor did we want them needing to learn or use the proprietary software interface.

We also developed custom input controllers (instead of requiring use of an on-computer GUI interface) to support situated input on the machine itself and improve the directness of interaction. On the quilting machine, this interface is mounted underneath the quilt surface, surrounding the throat plate. This allows players to push down on the quilt itself to guide the motion of the needle. On the embroidery machine, the controls are mounted to the side of the sewing area, due to space constraints; however, this control panel still supports players directly interacting on the machine (see in Figure 4.10).

4.3.4 Exhibitions and Press

Figure 4.16: Two players proud of their finished game of *Threadsteading* at Ctrl-Alt-GDC 2016 [70]

Threadsteading has been demoed at CHI 2016[10], Ctrl-Alt-GDC 2016 [70], the Sammy Showcase 2016 [200], and IndieCade 2016 [89]. At each of these exhibitions, the game was presented on the Singer CE-200 embroidery machine. One player (generally the winner) was able to take home the embroidered game board that resulted from their play experience (Figure 4.16):

Reception

The general response was positive: a mixture of curiosity and awe. Players were mesmerized by the machine's rhythmic chugging and the steady stream of stitches that revealed the result of the turn they took. The time between turns was also often used as an intentional game phase set aside for considering a player's next turn, so there was very little down time in between the bursts of sewing. The lack of boredom even during repeated stitch sessions demonstrated the user's ludic engagement with the machine and the game.

At the Game Developer's Conference (GDC, where Ctrl-Alt-GDC [70] was held), many visitors to our installation commented that they never thought they would see a *sewing machine* at GDC. In fact, at GDC the sewing machine's existence seemed to greatly distract from the visitor's ability to process the game itself. The culture of GDC is much different than those at other venues we demoed at, as it includes game enthusiasts, game academics, game developers, and representatives from the game industry, with a heavily masculine population bias. However, they still enjoyed themselves overall (Figure 4.17).

GDC also provided the most amount of press. Gamasutra [47], Rock Paper Shotgun [213], and Kill Screen [125] wrote articles giving an overview of the game. Gamasutra's article transcribes an interview with Gillian Smith. In addition, Make: interviewed Lea Albaugh and posted it on YouTube[7] [118]. Digital Design and Innovation also hosted an interview with me describing *Threadsteading*, which was posted on YouTube [60].

[7] YouTube [115] is a video hosting service.

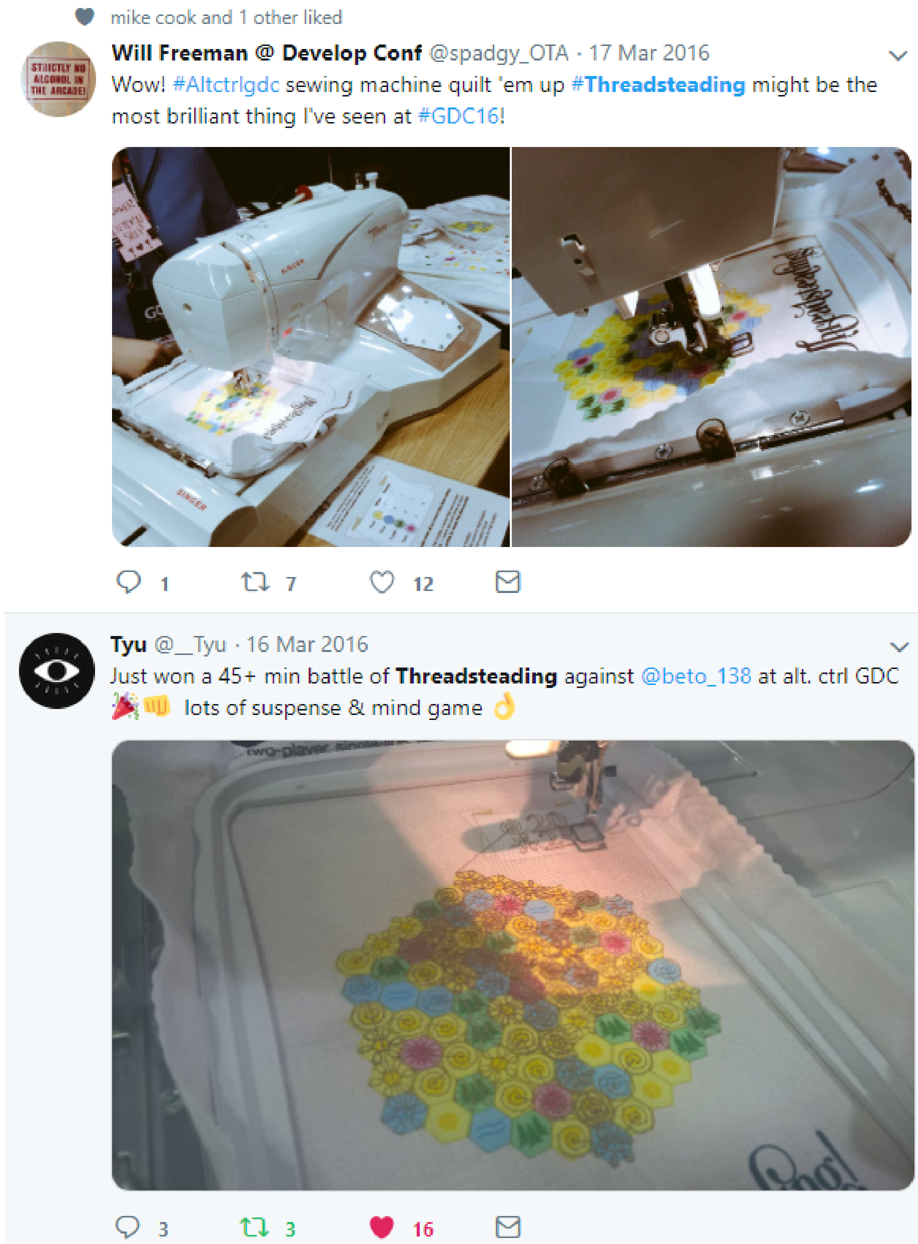


Figure 4.17: Twitter remarks from Ctrl-Alt-GDC attendees praising *Threadsteading*.

Indiecade 2016 was the other most notable exhibition. Gillian Smith and I encountered our favorite attendee: a young girl, about seven, who told us that she recognized the sewing machine from her mom and the Raspberry Pi[8] from

[8]The Rasperry Pi is a single-board computer that was used to handle the game logic and

her dad. She then asked her mom if she could make a sewing machine game too. We felt that we had accomplished something special by reaching out to future generations of game designers and crafters alike. Inspiring people by increasing their feelings and knowledge of agency — their confidence that they could do something like *Threadsteading* — was my favorite type of feedback.

The positive responses from attendees of IndieCade continued (Figure 4.18).

At the end of IndieCade 2016, it was revealed that our *Threadsteading* game won the Best Technology Award [90].

4.3.5 *Threadsteading* Lessons

In regards to the game design, there was a flaw in the design of our rules that we anticipated but chose not to restrict. We had hoped that negating the points for multiple traversals over the same territory and preventing immediate backwards exploration to negate the opponent's last turn would heavily discourage players from going over the same tiles more than once. However, some players were extremely competitive and adopted a "scorched Earth" policy, focusing on destroying their opponent's land rather than exploring. Because there were no rules to forbid this kind of behavior (for example, making it illegal to traverse a spot more than three times), games could go on infinitely as players danced around each other (Figure 4.19). In some cases, we intervened to call an end to these games to allow others to play, and in other cases the machine failed due to the fabric or threads losing structural integrity. We had considered rules to prevent this, but we could not come to a consensus on a simple set of rules that would not potentially leave the players in an unwinnable game state without a drastic overhaul.

direct both the sewing machine and the user interface.

Figure 4.18: Twitter remarks from IndieCade 2016 *Threadsteading*.

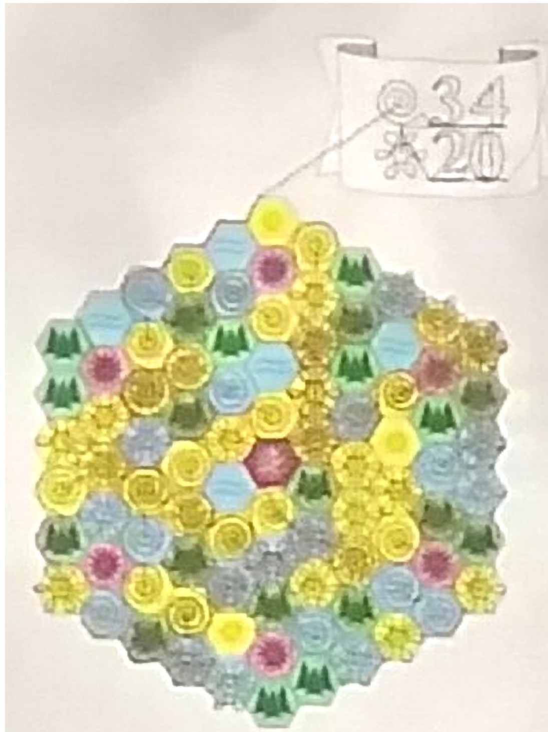


Figure 4.19: Two players played for nearly an hour during a period of low traffic at Ctrl-Alt-GDC and traversed nearly every hex in the map. There are 90 traversable hexes with a total of 97 possible points on this board, and only 54 points were scored across them. Tyu's board in Figure 4.18 is similarly overloaded.

In terms of ludic engagement, users were active throughout play, even when the sewing machine was chugging for over a minute between some turns. Users were constantly evaluating the board state, building knowledge of agency for the game mechanics as they would make a move and the machine would operate only within legal game actions. The low amount of options per turn — one decision with up to six directions — executed with a button press allowed people to engage with the sewing of the game immediately, even if they didn't understand the game.

4.4 Textile Craft Manufacture Insights

Every single project discussed in this chapter had a unique relationship with the physical materials and fabrication technologies that made up their domain, be it knitting, felting, sewing, quilting, weaving, beading, or embroidery. Not all of the projects explicitly incorporated game design, but they did involve active participation, pattern interpretation, and a playful and open-minded approach to the materials in the projects via ludic engagement. Sharing the design choices and/or manufacturing tasks with a machine may reduce agency due to a lack of confidence in a high-quality product, but it may also increase agency due to the lightened burden of crafting domain and manufacturing mastery.

4.4.1 Active Participation

In this chapter, I have examined works that involve either human crafting or human-guided-machine crafting. In all cases of the projects discussed in this chapter, the participant has never been truly passive: even players of *Threadsteading* are thinking of their next move while they are waiting for the game to sew the previous turn, and players of *Loominary* (see section 4.1.3) and *BeadED* (see section 4.1.3) were busy manually weaving or assembling their choices. Knitting for Spyn (see section 4.1.2) or Yarn Quest 2017 (see section 4.1.3) was over a longer stretch of time, but still required participants to regularly pick up their knitting. In both of these knitting projects, additional participation was also required in the form of recording media and documenting their use of Spyn, as well as making characters and playing the RPG in order to get their knitting charts for Yarn Quest.

Threadsteading

In section 4.3.3, I discussed how I spent the time to reverse-engineer the machines' protocols to make the game interactions as seamless and quickly executed as possible. From the beginning, the design team envisioned the game being played in an exhibition space, so we needed to account for people coming and going, for games to be started and ended arbitrarily, and for quick iteration of the game board. For visitors that wanted to stay for an entire legitimate game, we had to make sure it could be finished in a reasonable amount of time.

Figure 4.20: The template alignment tool we created for accurate hooping of the *Threadsteading* printed fabric.

However, one of our primary concerns was the game board and how to accurately make moves on it in order to make a recognizable and aesthetically pleasing result. We were not able to get any consistent and accurate registration — the actual vs. intended alignment of stitches — of the quilting machine's sewing onto the fabric that was mounted on the frame. The whole game board had to be sewn first, which took at least twenty-five minutes, in order for the player's motifs to fit within the game board's hexes. After being sewn, we had to manually mark the game board on the quilt with stamps before beginning, which is represented in the quilt you see in Figures 4.9, 4.11, 4.12, and 4.13. Luckily these troubles were avoided because the long-arm quilting machine was completely infeasible to use as an exhibition machine, so we came up with the alternative embroidery machine.

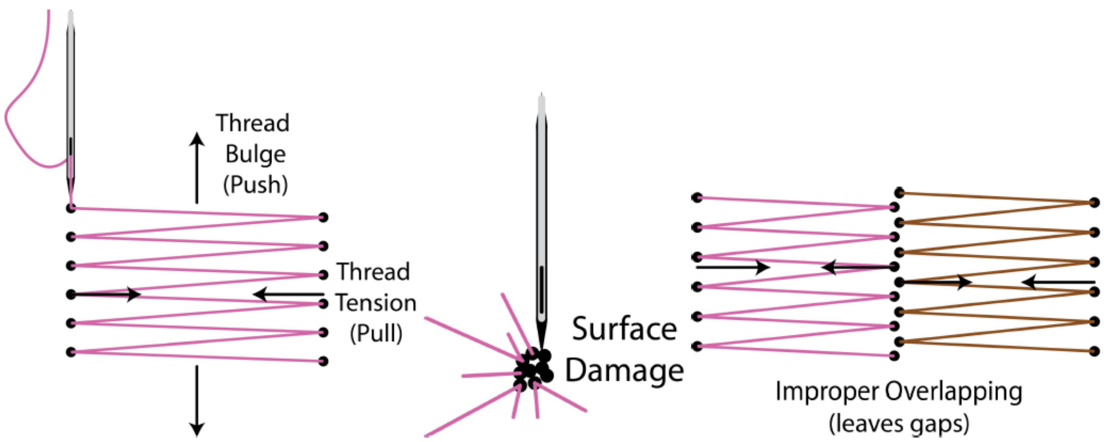


Figure 4.21: An overview of the most common physical sewing forces that negatively affect embroidery, usually machine embroidery. Left: threads pull inward parallel to their stitch direction (thread tension), while embroidery that bulks up pushes outward perpendicular to the stitch direction (thread bulge); Middle: dense embroidery in the same location will damage fibres, which can lead to holes and tangled threads that destroy the fabric or weaken its integrity; Right: the thread tension from the left image can cause a gap in sewing when two adjacent sewing areas pull away from each other.

We still had to figure out how to manage the game board. Sewing the board for each game on the embroidery machine took less time, roughly fifteen minutes, but still would have nearly doubled the length of an average game. However, getting registration on existing fabric in a hand-held embroidery hoop was a much more attainable goal. We created an acrylic template that would sit on the hoop as we attached the fabric (Figure 4.20). By aiming the center hex as closely as possible to the center hole in the template, and by aligning the whole hex board to the outline etched into the acrylic, we would get decent registration. However, there are always factors that could lead the game astray to watch out for, such as uneven tension, uneven stitching, or fabric damage (see Figure 4.21).

4.4.2 Pattern Interpretation and Agency

Textile manufacturing machines and the computers that control them simply follow instructions. Unless programmed to do so, these machines will execute commands that are useless, that damage their textile output, or that damage their own hardware (examples of these acts in the knitting machine, and how they were coded against in the knitting machine compiler, can be seen in section 3.3.4). While I may appropriate their manufacturing technologies, such as hijacking an embroidery machine to sew interactively with *Threadsteading*, how those technologies work is often not in the realm or skill set of textile crafters.

However, humans are creative and flexible when it comes to reading and adapting patterns in many ways that textile manufacturing machines are not (yet). Many crafters know their crafts deeply, closer to the systems-level knowledge discussed in section 3.4.1. Those participants that engaged with SkyKnit (section 4.1.1 knew their knitting would very likely end up being a mess, and the patterns could be completely illogical without adapting the source material. The user bevbh, a participant in SkyKnit, stated: “’I agree that an analogy with 19th-century knitting patterns is quite fitting...Those patterns were often cryptic by our standards. Interpretation was expected”’ [117].

Pattern interpretation also extends to surface-level design decisions. Some designs may suggest general colors, like Hoopla’s colored pixels (Figire 4.5, or the sample colors for embroidery patterns like those in Figures 4.6 and 4.7. In executing the pattern, there is no restriction on what threads or colors to use.

Many textile pattern materials are simply suggestions, unless otherwise stated, such as needing wool for the best results in needle felting. In fact, it is exceedingly rare – and may even be impossible — that a person following a weaving, knit, or crochet pattern use the exact same yarn, or for a quilter to use the exact same fabrics or colors in a quilt pattern. Material substitution can be simple or complex depending on the situation, but it is one of the first exercises in alternative pattern interpretation that a crafter will often exercise. Crafters also carry skills between different domains. Color theory, for example, drives material choices in all of the textile crafts mentioned in this dissertation – even blackwork embroidery is sewn in other colors.

It is important to note that pattern interpretation is a skill. Not only do novice crafters have difficulty grasping the alternatives within their domain, but many people are also simply not adventurous or confident in themselves to stray from given instructions. With much easier access to pattern authors and huge resources of other crafters online, many crafters outsource their problem-solving task instead. There are hundreds, if not thousands, of forum posts on Craftser asking for help finding and interpreting patterns tailored to their needs [15]. Some authors of free patterns with a public facing webpage, such as Nerdigurumi, have even stopped making patterns due to the behavior of their audience:

> I'm growing weary of the ongoing requests for assistance, many of which ask questions already answered in previous comments, are based on misunderstood instructions for which there are tutorials, or are people obsessing about tiny details that are not important rather than trying to figure it out on their own with trial and error. In addition many of these requests can be a bit rude, assigning blame for confusing directions or stating things like "this makes no sense, it's not explained at all!" or my favorite "I can't understand the pattern, can you make a video?" which translates to *This is hard so I don't want to try to figure it out, can you spend about 5 hours doing a bunch of work making a video so I don't have to bother trying to read instructions?* [136]

Pattern interpretation and improvisation requires increased knowledge of the domain to understand what design options can be modified and how to modify them. The designer/crafter needs to have confidence that the pattern will still result in a successful manufactured product after their edits. Pattern interpretation and improvisation thus require a high sense of agency with the crafting domain.

4.4.3 Playful Approach and Ludic Engagement

A few of the projects discussed in this chapter were designed to be games and successfully encouraged people to literally *play* them. These games utilized familiar genres and previous games: *Settlers of Catan* [196] for *Threadsteading*, choose-your-own-adventure narrative games for *Loominary* and *BeadED*, and tabletop adventure games for YarnQuest 2017. Making these craft games closely resemble or incorporate other familiar styles of games helped bridge the gap between crafting communities and gaming communities. The heavily male, game-oriented crowds at Ctrl-Alt-GDC and IndieCade were exposed to the physicality of crafts and the game design affordances of textile materials. Their general high agency with standard games brought down barriers to ludic engagement. Likewise, many of the female companions of these game aficionados, especially those that seemed bored and disengaged around the more traditional games, were intrigued by the creative possibilities of the mediums some of them were already familiar with.

The choices made in these games do more than change the game: they change the resulting artifact. All players of these games also encountered a duality in their choices during play: do I make my usual choice, or do I make the choice that makes the most beautiful artifact by the end of play? Choice in games is a complex topic in an of itself [123], where players of *Threadsteading* may play their turns to win at all costs, but players of *Loominary* may find choices more interesting when roleplaying particular roles. Some of the output from Yarn Quest 2017 had suspiciously balanced and organized results that lead me to suspect they were not necessarily from straightforward play sessions (Figure 4.22) What result counts as 'winning' or an optimal choice is very different for the different textile craft games discussed here. Some would argue that the act of participating in a mystery knit-along is a social game of discovery that crafters play [192].

Figure 4.22: Yarnia — The Grade Quest Blanket MKAL, pattern by Tania Richter, knit by Paula [156]. Note the four centered designs organized as a clear focal point to the whole blanket.

There were players that approached *Threadsteading* that did not want to play competitively or with another person at all. These players wanted to make interesting embroidery paths or wander around the map as if they were hiking through the wilderness. These textile crafting games seemed to inherently have appeal to multiple audiences that not only take different gameplay roles, but crafting roles as well.

Textile craft experiments need not be explicitly games to encourage playfulness. Participants of SkyKnit in particular voluntarily took the whole experience with a massive grain of salt, knowing their output would probably be ludicrous or unviable but nevertheless continuing with their participation. SkyKnit not only required the pattern adaption skills discussed above, but was clearly anticipated and enjoyed, as evidenced by the almost 2000 posts in the main thread on LSG's group forum [1]).

During Spyn, participants picked their own patterns and often had multiple projects in progress simultaneously. While the context of Spyn was more a formal user study rather than a natural internet phenomenon, the way users were encouraged to log daily progress made the processes of even ripping out stitches more enjoyable [160]. Participants more actively engaged with the process of knitting, elevating it beyond the usual hobby. The crafting process, the experience of gift-giving, and the experience by the recipient were all notably enhanced by the Spyn system in a variety of ways, such as emotional impact, engagement, and appreciation [160]. Not all forms of play are strictly for serendipitous fun.

4.4.4 Limitations

Textile crafts have a grounding in the physical world, with an infinite variety of threads and fabrics to use as materials, which adds complications that non-textile and digital projects do not have. If the physical limitations are not respected, the textile portion of the project will simply fail, often in a catastrophic manner: if cotton yarn were used instead of wool in the textile 3D printing project, the felting needle could break and the output would have little-to-no structural integrity or simply disintegrate during production; when players of *Threadsteading* disregarded our advice about retreading the ground, the fabric sometimes became damaged beyond playability.

In some cases, users can be buffered against this kind of failure, or the potential failure may not interfere with the crafting process. For example, players were barred from making illegal moves or sewing off the game board during *Threadsteading*, and if the player's weaving tension were too tight or too loose while playing *Loominary*, or if they missed a warp or weft in the weave, the RFID tags would still work just fine. The domain of machine embroidery is simply more prone to many more user mistakes than manual weaving on a rigid-heddle loom.

For *Threadsteading*, I spoke of how the rules of thumb in embroidery and quilting drastically narrowed the design space for our game: sewing a single connected line, sewing in one color, discouraging too much sewing in one location, and starting from the center of the fabric (section 4.3.2). Finding a game design that was actively engaging for both players, that was able to sew recognizable and intentional paths related to the game, and that followed these rules of thumb as closely as possible was extremely challenging. Sewing in a single line also made the blackwork embroidery project heavily favor designs that were a wholly connected graph.

Due to our extensive knowledge of machine embroidery, quilting, and their potential pitfalls, we were able to design a reasonably robust game that rarely encountered catastrophic failure. Due to the representation of the game on accompanying custom hardware, users were unable to cheat or subvert *Threadsteading*. The sewing machine's natural inclination to immediately pick up where the embroidery left off if a thread break or error occurred also meant that any disruption of gameplay could almost always be seamlessly continued. The only real conceptual flaw that would regularly lead to issues was the scortched earth player approach one discussed in section 4.3.5.

This chapter explored co-operative textile manufacturing across different crafting domains and using different kinds of commercial and custom technology. Framing co-operation with technology as a game invites curiosity and a willingness to explore these unknown experiences, both as a designer and a player, which is evidence of high ludic engagement. These experiences often also trigger both physical and digital types of agency, as well as disparate types of agency across different demographics, for designers and players.

Designers of manufacturing experiences have similar agency and ludic engagement as digital designers from the last chapter. Their physical agency over their craft and their digital agency with the machine interfaces both come to bear. Players of games are typically male and have high agency when it comes to standard game interfaces and game rulesets. Crafters are typically female and have high agency with crafting tools and materials, especially those in familiar crafting domains. *Threadsteading*, Hoopla, and *Loominary* experiences are simultaneously familiar and strange for both groups, and both groups increase their agency with the other's domains as they play. The following chapter continues to examine both designers and users of electronic textile products, which directly integrate computation and crafts.

Chapter 5

Perspective 3: Textile Craft Product

This chapter will explore how computation intersects with textile craft products and their audience, due to how I collapsed Rhodes' Product and Press into the Product (section 1.2) [153]. In my experience, computational textile output's biggest and most interesting challenges to agency and ludic engagement exist in the design and manufacture of them as well. Because of these two caveats of mixing product, press, as well as all three major crafting processes, this third and final large Perspective chapter builds upon the experience and lessons of the previous two Perspective chapters. This chapter will give an overview of the contexts in which computational textile output are made and used, will present my contributions in building two computational textile artifacts, and will summarize the challenges and design insights gained while building those artifacts. These insights will show how building computational textile artifacts requires additional domain knowledge outside of standard textile crafting practices, which makes crafters' agency generally low. However, the artifacts themselves usually take the form of familiar objects (clothing, jewelry, game controllers, stuffed animals, and toys), which entice and encourage ludic interactions and high agency from all ages and ethnicities. The primary community that the research in this chapter contributes to is that of HCI.

Traditionally, textile artifacts in and of themselves rarely have any connection to computational topics. However, with the rise of fabrication technologies and hobby electronics becoming more common, integrating those electronics with textiles has also become more popular. Physical textile craft output mixed with any form of electronic technology is commonly referred to as electronic textiles (e-textiles) or soft circuits. This chapter will be focusing on hobby and small-scale e-textiles made for playful, explorative, and educational purposes.

All forms of textile crafts mentioned in this dissertation can be integrated with electronics. Conductive materials such as carbon, silver, copper, nickel, and steel are used to coat or are made into raw fibres, spun into threads and yarns, and woven or knitted into fabric. There are countless applications of e-textiles that prioritize different attributes of the materials, although the most common is tarnish resistance of metals that oxidize. For this reason, many materials are not pure metal, and there are hundreds of blends of materials sold by different manufacturers. The most common non-fabric materials sold by hobby electronics stores like Adafruit [7] (Figure 5.1) and SparkFun [183] are made of stainless steel. It is very difficult for pure metal threads to be used on industrial weaving and knitting machines, so they are often plated or have conductive and non-conductive blends instead.

5.1 E-Textiles Research Review

Electronic textiles have been booming in the past decade thanks to the democratization of technology [194] and the prevalence of simple kits that are marketed

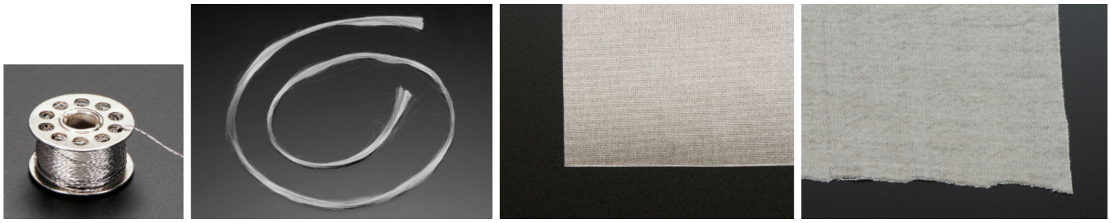


Figure 5.1: An assortment of conductive materials sold by Adafruit, all made of various metals with different properties. From left to right: 2-ply thread [5], loose fibres [3], woven fabric [6], and knit fabric.[4].

to people as young as 7. Buechley et. al. included children as young as 8 in their electronic textiles workshops [30], and electronics books targeted at children aim for 7-10 at the youngest ages (as listed on Amazon[1] product pages for the following: [58, 172, 108]). These introductory projects tend to focus on the understanding of simple circuits, but often also include projects such as pianos and other interactive toys.

Maker spaces have also been becoming much more common across different institutions and independent workshops [119, 59]. However, these maker spaces have a strong bias towards electronics; soldering stations and electronic components are practically ubiquitous, but sewing, embroidery, and serger machines are much rarer. Leah Buechley has suggested entreating craft communities for their input on tools and materials in order to encourage their participation in these maker spaces [29].

As a quick aside, I will briefly mention Spyn here as an instance of a computational textile output that is relevant to this chapter as well [159, 160]. Spyn was examined in-depth in Perspective 2, section 4.1.2. While there are no physical electronics tied to Spyn in its second study, it was electronically annotated with a digital representation of the physical craft with accompanying media. It is interesting to note that the more advanced physical computing elements of the

[1] Amazon.com is an e-commerce website [13].

first iteration of the project were removed and redesigned due to their clunkiness and impermanence.

5.1.1 Educational Kits

People learn by making and doing. Both textiles and electronics have instructional kits for all kinds of domains, and it is unsurprising that there are kits that aim to enable both skills via e-textiles. These simple educational kits lower the otherwise high barrier to entry for crafters to grasp the electrical engineering and programming concepts related to e-textiles. A lower barrier to entry leads to a more confident feeling of physical agency and a higher likelihood of enjoying the e-textile crafting process via ludic engagement. In particular, the more familiar sewing interface to the LilyPad Arduino has attracted more women to e-textiles [24].

MaKey MaKey

While not strictly tied to textiles, the MaKey MaKey [177] is famous for it being the simplest of electronics kits that also includes integration into textiles. Due do how the MaKey MaKey interprets electrical resistance, it allows for many more forms of conductive materials than most other electronics. Thus, a wide range of materials such as “the human body ... food, plants’ and soil” work as input devices. These objects trigger keyboard and mouse events so the MaKey MaKey can work with any software that takes those inputs. The MaKey MaKey has been used with participants ranging from children (as young as 10) [107] to the elderly (as old as late 80’s) [158] to make and play with custom controllers made of Play-Doh[2] and fruit respectively.

[2] A proprietary modeling compound used by children. It is conductive enough to be used as functional components of the MaKey MaKey.

The challenge of using the MaKey MaKey for e-textiles is that most fabric is not inherently conductive. However, conductive textile materials, like those shown in Figure 5.1, can be used as plug-and-play input devices to the MaKey MaKey without sewing. Its ease of input to other programs makes the MaKey MaKey extremely attractive for simple electronics projects. For example, *BeadED* was made using a MaKey MaKey for its bead selection detection (see section 4.1.3) [190].

LilyPad Arduino

The LilyPad Arduino is designed as "a fabric-based construction kit that enables novices to design and build their own soft wearables and other textile artifacts" [29]. Unlike the MaKey MaKey, the computational center of the LilyPad is a microcontroller that is intended to be programmed. Unlike other microcontrollers, however, the LilyPad is intended to not require any soldering. The design of the microcontroller is on a circle, which was found to be more space-efficient and easier to connect lines of thread without short circuits than the standard rectangular shape for most boards [29]. Around the perimeter of the circle are tabs of conductive fabric that connect to pins on the microcontroller, including power, ground, and many general-purpose input/output (I/O) ports. Users are intended to sew through the tabs with conductive thread and program the tabs via the Arduino IDE[3]. The visual history of LilyPad iterations are shown in Figure 5.2.

[29] presents further user studies on a soft iteration of the LilyPad, where children ages 10-13 successfully used the kit to build e-textiles. However, the programming was challenging for many participants, where a few of the participants explicitly stated they would not continue with anything related to programming due to its difficulty. It is important to acknowledge that, while the

[3] An open-source programming interface most commonly used with Arduino hardware.

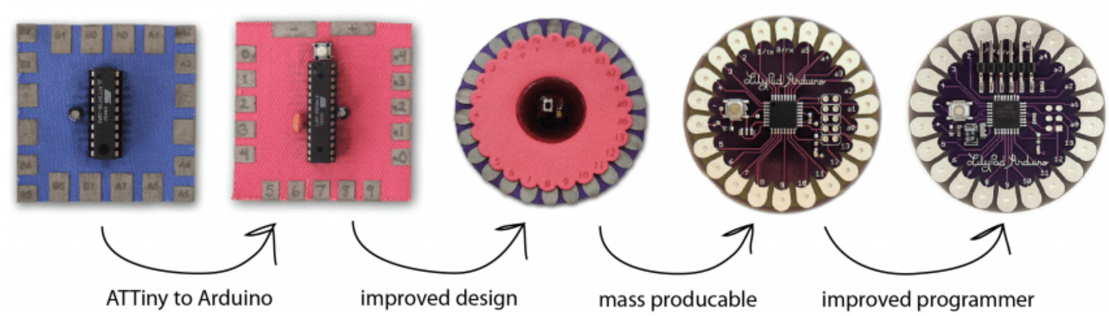


Figure 5.2: The history of the main LilyPad Arduino microcontroller [8]. The first image on the left show iterations presented in [30]. The center image is the iteration shown in [29]. The right image is the most recent LilyPad Arduino 328 Main Board.

LilyPad Arduino does reach a broader audience for e-textiles than most other microcontrollers, users may still lack ludic engagement and a feeling of agency toward making e-textiles.

After the project iterations presented in the 2008 paper [29], Buechley paired up with SparkFun's Nate Seidle for mass production. Now there are dozens of kits and packages of common sensors and actuators for a multitude of projects, such as switches, buzzers, light sensors, accelerometers and support for bluetooth [182].

5.1.2 Wearables

Technological devices meant to be worn somewhere on our bodies are considered wearables, and much of what we wear is a form of textile. The commercial industries have been examining wearable technologies for personal devices such as smartwatches that can interface with your mobile device, as well as many forms of fitness tracking via heart monitors, temperature sensors, sweat sensors, and embedded step counters in shoes. Google has invested in the Google Glasses project [75], as well as project Jacquard that would integrate a touch interface into the surface of denim jeans [76]. Wearables are effective for tracking certain medical conditions, such as foot monitoring socks for diabetic ulcers [178].

The example studies done with the LilyPad Arduino discussed in the previous section often resulted in wearables, including a hat, bag, shirt, and sweatshirt in [29], as well as shoes and a bracelet in [28]. Other papers have run studies exploring the educational potential of the LilyPad wearable design and creation [167, 94, 168]. While these studies focused on the prototyping process and mostly simple designs, there has been additional research exploring wearables for games, social functions, and self monitoring.

Wearables for Games, Play, and Social Interaction

Some research has been done exploring not just generic tests of e-textiles education, but of focused explorations of the MaKey MaKey and LilyPad Arduino, along with the Scratch and ModKit coding environments, to make wearable and non-wearable game controllers [154]. Examples of the wearable artifacts made by participants of that study include a partial glove controller for a *Flappy Bird*[4]-style game and a jousting game using conductive fabric vests. The HCI/UX[5] communities are becoming increasingly interested in forms of interactions via wearable e-textiles that are outside of the direct applications of sensors (e.g., sweat detection via conductivity/ph detectors; step counters via accelerometers, pressure sensors, and/or GPS, etc.).

[4] A one-button obstacle avoidance game, where the user fights against gravity and physics to navigate precise areas of safety. The original game, *Flappy Bird* was created by solo game designer Dong Nguyen [137].

[5] User eXperience.

5.1.3 Social Wearables

Multiplayer games and play revolve around social interaction as a key part of their design goals, such as the jousting game in [154]. Another competitive physical wearable game is *Swordfight*, where players use strap-on harnesses equipped with Atari 2600 controllers to joust with their crotches and hit the opponent's action button before theirs gets hit [135]. Social wearables also exist outside of the context of games, such as being used as group discussion monitoring devices and as hug detectors [141].

A collection of games research related to wearables has been focused on augmented LARP (live-action-role-play) experiences [207, 56]. An in-depth synthesis of the research from those projects is presented in [57]. The design of these LARP support devices required players to interact by physically touching or connecting to different parts of the devices. Interdependence using one-sided information and devices only accessible by other people (such as on the participant's backs) is a means of forcing communication and interaction between players. E-textiles add a dimension of embodied systems that often involves collocated interaction to make use of their physicality. The following projects are e-textile social wearables that focus on games and playfulness between their players.

EmRoll

An example cooperative e-textile game controller is *EmRoll*, where two players don suits with different colored hoods that are embedded with sensors in their legs and arms [220]. Players of *EmRoll* have to dance to portray different emotions, such as happy/excited or relaxed. While direct physical contact is unnecessary, players often joked and encouraged each other with physical involvement, which made the multiplayer experience far more emotional and impactful. However, detecting emotions through various sensors, such as GSR[6] for fear, was inaccurate, especially due to the physical activity of the game's mechanics. Failure of other sensors, such as getting appropriate breathing timing, or detecting jumping up and down, would fatigue or cause other physical distress to the player that disrupted play. Failure of these types of sensors is a common occurrence, see section 5.3.1 below for more detail.

Hotaru

Isbister et.al. presented the design practice of *interdependent wearables (for play)* in [91] as a culmination of insights during the creation and exhibitions of their cooperative project, *Hotaru, the Lightning Bug Game*. One player has a gauntlet that can project energy into the sky, while the other has an energy 'tank' that powers the other player's gauntlet, and both players have gloves with embedded sensors. The player with the tank gathers energy using gestures, and when it's full, the player with the gauntlet reads the tank's energy, initiates holding hands to transfer the energy, and releases the energy with their gauntlet's gesture. The goal is to complete this series of gathering and releasing energy as many times as possible in a short time frame.

In order to complete the game, both players must perform gestures, hold hands, and communicate, even at a non-verbal level. Of the 66 players, 73% indicated that their trust, liking, and connection with their partner player was influenced.

5.1.4 Non-Wearables

Continuing from previous wearables section, this section examines non-wearable e-textile artifacts that contain a focus on the core experiential goals of the artifact.

[6]Galvanic Skin Response. GSR sensors are one means of measuring sweat gland activity.

While the following projects do report on their design philosophy and processes, the end products of the research exemplify those experiential goals as the primary demonstration of their research contributions.

***eBee*, an E-Textile Board Game**

A new venue for playful e-textiles is the incorporation into board games, a natural physical venue for the injection of physical crafts [33]. External game rules and other physical pieces related to the game inject a context of meaning and purpose to the play involving e-textiles. Simple circuits also thrive in this environment, as it is up to the player to setup and make use of the circuits, and they should not distract from other game design elements. Similar to the properties and boons of *Threadsteading*'s design and presentation — intriguing women and men, children and the elderly, as well as different ethnicities (section 4.3.4) — craft games made with soft circuits have wide appeal and curious design restraints and affordances for game design.

eBee is a board game using soft circuits as its main mechanic and form of construction [33]. A mix of conductive velcro and conductive fabric allow the game hexes to make secure connections across pieces as they are placed on the board. The goal is to make a circuit between the central hub power source and LED island tiles. The players must respect the positive/negative flow of electricity for the LED to be lit. The final game is an elegant form of quilt style and electrical function: different styles of tiles use different fabrics, players use different colored fabric, and the LED tiles allow three directions of ingress to take advantage of the hex piece shapes.

The game was successful in its appeal across gender, age, and ethnicity groups, especially toward older women. When mistakes occurred in the electric polarity during play, it was a group effort to "debug" their circuits. Co-operation was also a common theme that emerged. Often after play, players would experiment with lighting more than one island, and making single long circuits between all the islands. It was clear that the basics of how circuits function was explored by many users that may not have been familiar with it at the beginning of play.

Fidgets

HCI practitioners and researchers have previously engaged in some exploration of the space of handheld fidget objects that have technological augmentation. For example, Mind Spheres [147] was a pair of wooden, LED-studded spheres that the user carefully rotated in one hand, similarly to Chinese Baoding balls. Moving the spheres in the right way would cause the lights to glow in particular patterns. The Relax! pen attempted to track user states such as frustration, and provide some kinesthetic feedback to help prompt them to modulate their state [12]. The Skweezee system provided tools for sensing squeezing of soft materials and software that could be used to capture patterns of use [206]. Perner-Wilson et al. conducted workshops in which participants constructed textile interfaces [143], and have provided toolkits and resources that Cottrell and I found helpful in our research for the next section [145]. Research groups have also explored and cataloged materials used in "Smart Material Interfaces" [127, 210], materials that can make expressive changes such as electrochromic or electroluminescent materials. No one that we are aware of has systematically explored the material space that we present in the following section on Fidget Widgets.

5.2 Fidget Widgets

This section (Fidget Widgets) is adapted from the previously published paper Soft-bodied Fidget Toys: A Materials Exploration *with accompanying authors Peter Cottrell and Katherine Isbister (for full citation, see [45]). My contribution was as a craft advisor, designer, and manufacturer in both the brainstorming and production phases.*

Figure 5.3: Final prototypes resulted in a 9“ x 5“ x 5“ hedgehog and a 12“ x 6“ x 4“ Dragon both larger than expected but still appropriate for children.

Presented in this section is an exploration of e-textile/soft materials that can be used to capture fidget traces, while providing the touch sensations that fidgeters report seeking out in everyday fidget objects [98, 99, 97]. We created two soft-bodied “sampler” objects with a range of smart fidgeting affordances, which we describe in this section, along with a general outline of the range of properties explored. This work extends exploration of a novel design space introduced in [99], toward the end goal of creating smart fidget objects that aid self-regulation. We include an overview of our design process, present some preliminary insights about materials that support this design space, and conclude with current and future directions for the work.

Fidgeting is a natural and habitual practice that people seem to use to distract themselves and focus their minds [36, 98, 97]. For example, people keep objects nearby for fidgeting when they are doing deskwork. A study of fidget objects [99] collected using an online Tumblr gathered examples that ranged from ready-to-hand items such as pens and paper clips, to toys such as spinning tops, to desk ornaments like Newton's Cradle. It has been suggested that capturing touch traces could be beneficial in aiding fidgeters' attention management efforts — helping a person to realize when they are fidgeting a lot, and might want to take a break or shift tasks, for example [99].

The present work builds upon prior study of objects that people report fidgeting with [99], focusing on particular sensations that fidgeters are attracted to that are not provided by current commercial technology products, and looking to generate those kinds of sensations with soft electronic/e-textile materials on a soft-bodied object. We present two "sampler" objects that we created, which offer the user multiple modalities for interaction. In addition, we describe an underlying technological infrastructure we created that allows capture and storage of touch traces.

5.2.1 Fidget Widget Research Progression

First we performed a materials exploration, grounded with a list of touch qualities drawn from [99]'s study of self-reported fidget objects. These included "crinkly, squishy, squash, snap, hissing, strumming, clicky-clackety, cool, smooth, rough, mush, twirl, spin, roll, bounce, shaking, braid, flipping, clicking, scrunch, squeeze, rub, and twiddle." Examining the objects and descriptors from this study led us to conclude that fidgeters seek a combination of complex sensations in fidget items, one that is not broadly available in today's technologies [98].

We set out to develop prototypes of objects that encompassed as many of these sensations as possible, which could also capture touch traces unobtrusively. **It was my task to embed conductive materials within crafted fidget objects**, including objects that were knit, crocheted, and sewn, as well as some non-textile crafting approaches such as polymer clay sculpting. These objects, when fidgeted with, should create an electrical signal distinct from when they are not being fidgeted with, in order to capture touch traces unobtrusively. I encountered first-hand crafting difficulties as a designer and manufacturer of these e-textile prototypes, as I had very little foundational knowledge with electrical engineering and microcontroller programming. I required Cottrell's guidance in order to understand what would and would not work in terms of managing battery power and ensuring safe interactions with our target audience. For example, having the default state be a disconnected circuit and the fidgetted state be a connected circuit minimized load on the battery and maximized the toy's longevity.

An additional constraint of our explorations was an emerging partnership with researchers interested in supporting self-regulation behaviors in children. We decided to focus our design efforts on crafting objects that could potentially appeal to children ages 8-10, to prepare us to engage with this future use case. In a companion study, we explored how fidgeting in children presents many of the same heuristics of interaction that it does for adults as reported in [99]. This companion study that makes use of the fidget samples is discussed below in section 5.2.4.

We gathered two sets of materials, those that mimicked or had some traceable properties and those that would form the infrastructure around which the smart material could be integrated. Notable materials included in our collection:

- 2 varieties of conductive fabric, MedTex180 (a soft knit variant) and RipStop (a stiff woven variant)
- Faux Leather to imitate Eeonyx non-woven resistive fabric
- Pressure sensors made of Velostat and conductive tape
- Knitted yarn balls with interwoven conducive tread to make squeeze sensors.

After gathering our materials, we sketched out ideas that would make for interesting repetitive use interactions drawing from fidgeting actions as described in our criteria. We aimed to envision one unique interaction for each word with some basis in traceable data. Some examples include: “squash” a ball sensor to determine grip strength, “strum” a field of flex sensor embedded grass, and “rub” the belly of a creature with embedded conductive thread strands.

By the end of our brainstorming we had a number of ideas including a potted plant that had bunny ears instead of leaves, which would light the pot up when the ears were squished or played with, a hedgehog that would flex its spines when threatened and relax when its belly was rubbed, and a set of conductive scales that would understand when the creature was being reshaped. We finally settled on two main fidget-able plush toys that would incorporate most of our favored sensors — a hedgehog and a dragon. We built both sampler creatures in an iterative process that is described in Research Results.

5.2.2 Hardware Design Parameters

As part of our effort to make objects aimed at 8-10-year-old children, we had two key requirements in terms of hardware: 1) keep wires contained within the animal as much as possible, and 2) keep the objects small so as to fit the hands of children. The limiting factor in size would be the main microchips. We evaluated then-current-market-ready microcontrollers with the smallest possible footprint (to maintain minimum possible size) that would be able to communicate wirelessly (to keep wires contained within the toys while in use) and had an onboard battery that could last for multiple days. Rather than compact several smaller boards that each served one of these goals we found a microcontroller that had all three functions in one.

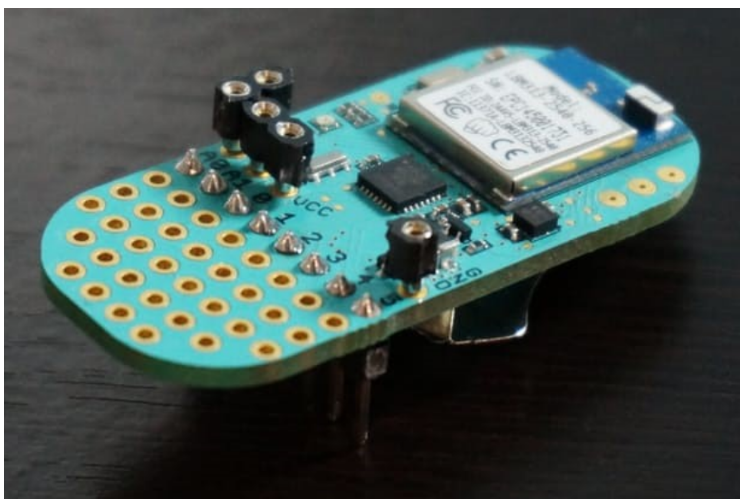


Figure 5.4: A LightBlue Bean microcontroller with interface sockets

The LightBlue Bean from PunchThrough (see Figure 5.4) is 3 inches long, and has onboard Bluetooth, protoboard and battery which makes it suited to soft structured objects such as fidgeting toys as it reduces the chances of the board getting crushed while being manipulated. Padded pockets embedded in the toys further protected the hardware during use and also enabled access to the battery. The Bean has 2 analog channels (capable of reading in a range of values from a sensor) and 6 digital channels (capable of detecting if a switch is open or closed), which means we needed to be strategic about how many and what style of sensors we tried to integrate into a single design. However, the reduced array of sensors allowed us to get an extended battery life allowing for minimal down time. Unfortunately, the LightBlue Bean uses single-use coin cell batteries rather than the more popular rechargeable lithium batteries, but with the Low-Energy Bluetooth protocols the battery lasts for multiple days of interactions before losing power. Programming was done in a mix of Arduino and Blue Bean-specific coding.

5.2.3 Culmination of Fidget Widget Research

Hedgehog

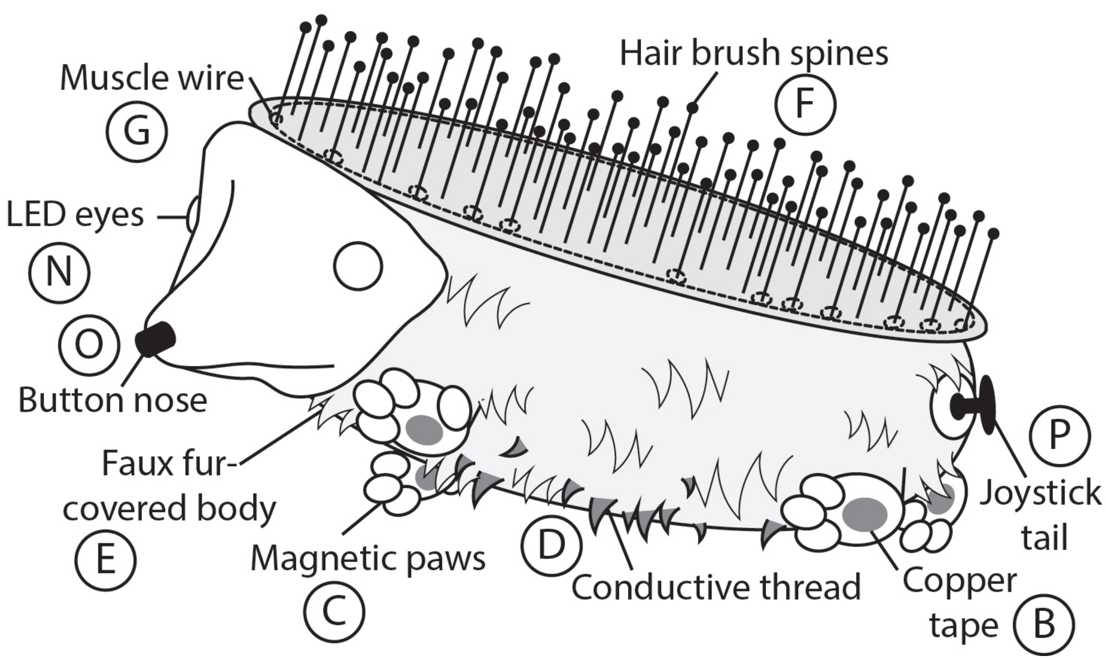


Figure 5.5: A diagram of the hedgehog.

Figure 5.5 is an illustration of the hedgehog's features (see Figure 5.3 for photo of the final prototype). Our two planned keystone sensors for the hedgehog were a pressure sensitive squeeze ball and a prickly sheet that could detect when a user is stroking it. Due to our target audience, we replaced the prickly sheet with a child-friendly metal-spinned hairbrush with plastic nubs on the tips. We ultimately removed the pressure sensing portion of the squeeze ball so that we didn't encourage users to apply pressure to the microcontroller, but kept the plush interior to invite some light squeezing.

We then added LED eyes for interaction feedback, a push-button nose for clicking, a joystick tail for rolling under the thumb, and a field of conductive threads embroidered into faux fur fabric to detect when a user is petting the belly of the beast. The conductive thread is embroidered in two separate sections; when stroked the conductive fibers flatten and create a loose connection detectable by the microcontroller.

We combined magnets, partially embedded copper tape and clay to form paired surfaces that either attract or repulse. With the exposed copper tape and paired magnets, a detectable connection for the microcontroller is created when paws meet (see Figure 5.6).

As we were assembling the various sensors we realized the hair brush had a flexible foundation that when morphed could present anthropomorphized feedback to the user. Attempting to create this capability, we added a ring of muscle wire around the perimeter of the hairbrush. However, when testing we realized the muscle wire either didn't have the leverage or the physical strength to stress the brush back to create the desired feedback. In future iterations of a similar prototype, this is a feature the team would like to revisit.

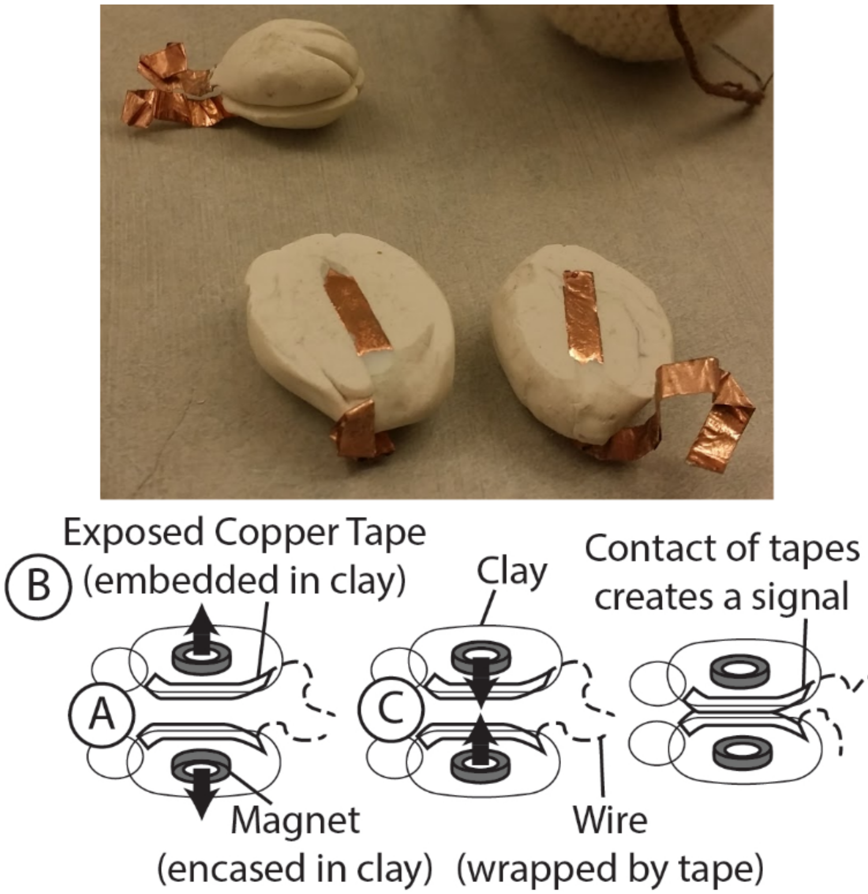


Figure 5.6: An initial prototype (top) and x-ray side view (bottom) of paws showing the construction of the embedded magnets and copper tape. Magnets are wrapped in polymer clay, and copper tape is anchored on the surface of the clay. The orientation of the magnets directs the polarity of the object. Clay, magnet, tape, and wire are baked per clay instructions to harden. The tape must be the most protruding surface to make contact. Thicker conductive tape on a protruding surface offers the best means of contact.

Dragon

Figure 5.7 presents an illustration of the dragon's features. The initial concept for the dragon revolved around a sheet of conductive scales that could detect when the scales shifted position and a conductive-fabric-lined squeezable tail filled with a variety of beads and stones to create a smooth texture within the tail. Lacking a reliable distributor of Eeonyx resistive fabric, we substituted faux leather for

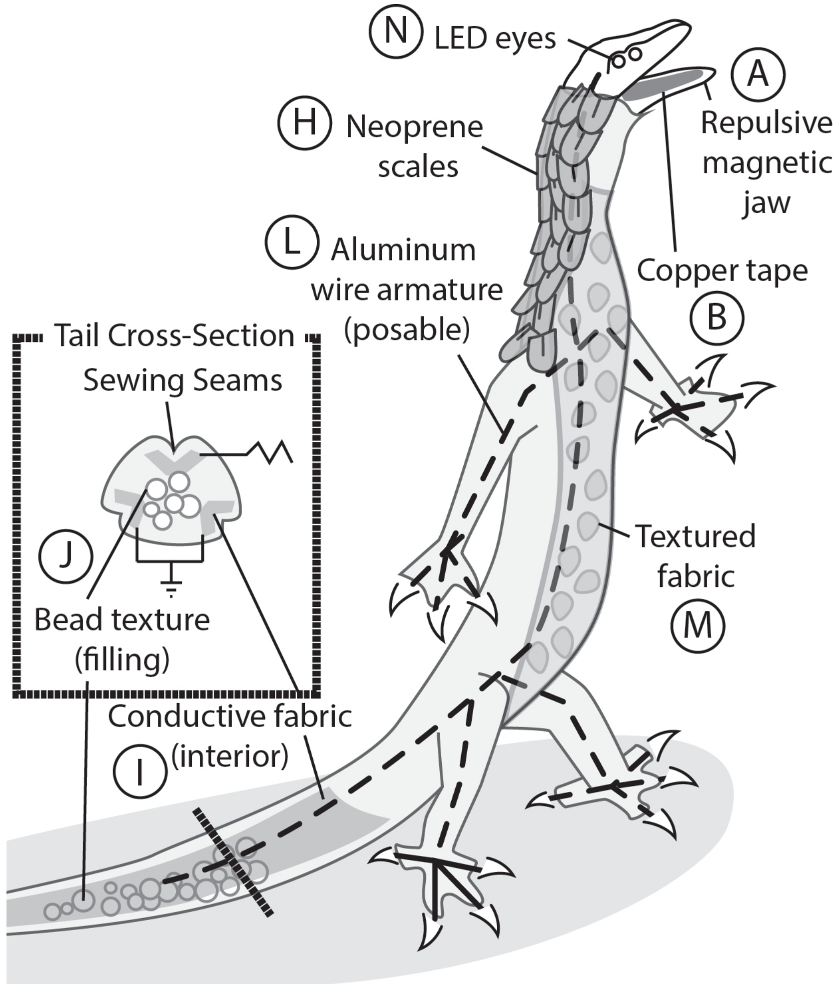


Figure 5.7: A diagram of the dragon. Letters correlate to smart technology embedded into the dragon.

the more substantive conductive material to provide the flexion we were looking for to differentiate it from the rest of the body. The conductive tail was the single keystone that maintained its initial design from brainstorming to implementation (see Figure 5.7). We started implementing the build with an aluminum frame around which we could layer different fabrics while retaining the gangly nature of a reptile, which made the dragon distinct from that of the plush hedgehog. We draped scales around the shoulders of the dragon (H), and found scale-textured faux fur that complemented the shoulder scales and used that for the underbelly.

We also implemented several similar sensors to those in the hedgehog; repellent magnet pairs, clay claws and LED eyes.

For the weighted squeeze sensor that acted as a tail, we lined the inner side of the passage with pieces of conductive fabric so that as the poly-fill, rice and marbles inside the tail are moved aside the two conductive pieces of fabric meet and create a circuit. This allows the microcontroller to detect approximately how much pressure the tail is being crushed with due to corresponding variable resistor values.

5.2.4 Fidget Widget Discussion

My experience crafting the prototypes and final exemplar animals required assistance from Cottrell, someone more familiar with electrical engineering principles. I had no training beyond understanding the difference between a connected and disconnected circuit. I had extremely low feelings and knowledge of agency with regard to how the circuits should operate when embedded in textiles. While the exploration had evidence of ludic engagement, I feel that is due to my disposition rather than representing a general rule. It took hours of iteration with Cottrell and investigation of tutorials online for this project to be at all possible for someone unfamiliar with electrical engineering, a problem I will further discuss below in section 5.3.1.

While building the exemplar animals, we initially aimed to implement multiple forms of sensors so that each sensor or feedback method was implemented uniquely. We found that what were perceived as unique implementations of sensors could ultimately be collapsed down into similar hardware design, but with different perceptions of the hardware based on the contextualization. For instance, the embedded magnets provided different experiential properties depending on whether the magnets attract or repel each other. This meant we could encourage users to bring the paws of the hedgehog together by creating paws that could barely reach due to their positioning on the body, but when they do connect they stick together. Whereas the dragon's mouth refused to stay closed (due to repulsive magnets) and when a user forced its mouth shut we could use the LED eyes to react.

In the process of assembling the exemplar animals, the authors and their labmates expressed similar interest in specific experiential qualities. The hairbrush, the filled squeeze sensor, the embroidered faux fur, and the repulsive magnets evoked unique and unexpected pleasure. To help understand why people drifted to those textures and interactions, we decided to catalog their experiential qualities here.

- The hairbrush's flexible back and prickly alert spines made it pleasant to strum, squeeze, morph and poke.
- The mixture of microbeads and rice against each other and the slippery texture of the conductive fabric made the squeeze sensor pleasant to squish, roll, and pinch, and had a satisfying weight that made it pleasant to swing around in its initially detached state.
- The faux fur belly was soothing with its soft fuzzy fluff.
- The magnet pair placed in the mouth of the dragon encouraged fiddling since the jaw was long enough for a comfortable thumb and forefinger to pinch easily, but with its floating drifting mid-air slip kept a pleasant resistance between your fingertips.

Study

After the completion of the fidget toys, they were included in a long-term study with 28 children between the ages of 6 and 11, as well as 24 parents and 2 teachers [55]. The goals of the study included examining children's fidgeting behaviors, determining if the children had different fidgeting preferences under different circumstances, and what purpose fidgeting may have had for the children. The parents' and teachers' presence helped give the researchers further insights on long-term behaviors, while the children were observed during five sessions, including one each of exploratory and brainstorming activities.

Fidget objects were supplied during the exploratory phase to isolate different fidget activities. The fidget toys designed and created in this chapter were presented in the brainstorming phase, along with their raw materials, as inspiration for the children to design their own ideal fidget items. The children drew diagrams similar to Figure 5.5 and Figure 5.7 for these items. Three of these items resembled animals like ours — platypus-like, cat-like, and cat-dragon creatures — as well as abstract items that assembled their favorite fidget activities. Reported top materials were things like soft rubber and plastic, while top fidget objects were squeeze balls and Orbeez[7].

How Can We Improve?

Our fidget designs had the goal of detecting touch activity unobtrusively (described in the section 5.2.1 above). In order to do this, we had to detect either a change in the connectedness of the circuit, or a change in the resistance values of a connected circuit. E-textiles offer an extra level of challenge in terms of robust circuits due to the flexibility of the materials and general lack of insulation

[7] A proprietary version of a superabsorbant polymer, able to absorb up to 99.9% liquid in order to become gel-like in consistency while still being able to hold its shape.

between conductive materials (discussed more below in section 5.3.1). The behaviors we gravitated towards were those easily detectable using discrete activities or ones already supported via hardware tools, such as making use of a button and a joystick on the hedgehog. However, these activities often did not align with current fidget activities that the children had a bias towards. For example, we had a fabric squeeze/manipulation sensor in the dragon's tail, but it felt nothing like the children's preferred squeezing materials of elastic, silicone/plastic, and fluid-filled toys.

The children's highest priorities were reported to be a discrete[8] toy that was readily at hand[9] and robust and/or disposable. A fidget toy should be robust and/or disposable because a majority of fidgetters reported aggressive fidgeting activities that actively tested the boundaries of their fidget toys. Our exemplar fidget objects arguably satisfied none of these particular priorities: the exemplar animals were larger than most fidget toys, not likely to be readily accessible due to this size, and considered by the children to be delicate. Indeed, the nature of the hand-made toys discouraged aggressive fidgeting that tested the endurance of the object, which a majority of fidgetters usually experiment with. None of the isolated fidget items explored within the study included circuitry like ours, which likely speaks to both the investment and care the children may have with electronics. For example, children likely do not consider Furbys[10] or other robot animals to be fidget items so much as companion pets.

Finally, while we focused on many different surface textures: hard and smooth clay, pointy claws, hard plastic button and joystick, soft fabric, fuzzy fabric, vel-

[8]Non-discrete toys could be taken away as being disruptive or distracting, or they would immediately expose the child's fidgeting activities.

[9]The act of unconscious fidgeting necessitates being able to grab and fiddle with an object without thinking about finding or using it.

[10]Proprietary robot toy pet of indistinguishable species with relatively advanced artificial intelligence.

vety fabric, slick faux leather, and metal magnets. The majority of catalogued fidget activities, other than stroking or petting, do not consider surface texture to make much of a difference. In addition, as stated above, the children generally preferred non-organic plastic and elastic sensations. The fuzzy tummy fabric of the hedgehog was particularly soft and attractive, however, and has since been carried on to yet unpublished future iterations of the fidget toys.

5.2.5 Fidget Widget Conclusions

Overall, the team was satisfied with the evocative set of soft electronics/e-textiles options generated in this research, toward building soft-bodied fidget toys, which can enable the kinds of sensory experiences that people want in fidgets while allowing for capturing touch traces. Every step in the production of this project, from ideation to construction, required a blend of electrical engineering and physical craft expertise. Electrical engineering provided the hard constraints for types of conductive materials and means of interaction that could produce a detectable change in analog signals. For example, while working, our crafter was looking for a sharp snap interaction and suggested solid metal snaps as would be used on overalls to create a solid connection, but was cautioned against exposing users to electric shock, and told that constant contact could easily drain the battery if used as part of a circuit. Physical craft expertise provided a library of tools, materials, and construction blueprints for embedding, disguising, and enhancing the hard constraints of electrical engineering. Without the expertise of the crafting team the engineering team would have created much rougher, simplified examples, and amalgamations like the conductive thread embedded belly would have been dismissed out of hand.

This style of prototyping heavily relies on the creators' skill sets. Well-documented smart material crafting tutorials can help improve understanding between teammates and when shared, also broaden the skillset of the community at large, as exemplified by blogs like that of Perner-Wilson and Satomi [145]. In addition, workshopping and skill share events that teach practical prototyping skills such as those hosted in the TEI Studio series greatly improve the overall skill and quality of the community. Quality work comes from not only have the right tools, but also the creators to use the tools, and it's our recommendation that smaller communities develop a series of skill share workshops when possible.

5.3 Textile Craft Product Insights

Throughout this chapter, we have explored a wide range of approaches to designing, making, and experiencing e-textiles that can be separate from the digital representations in Perspective 1 (chapter 3) and Perspective 2 (chapter 4). I have focused on an easier approach to e-textiles using simpler tools, interfaces, and design goals, from the related research topics to the Fidget Widgets project. However, even these relatively simple projects (with simple circuits) have come up against low agency problems. These problems are an expected symptom of the low likelihood that women with textile experience are also willing and able to become familiar, or at least get comfortable, with electrical engineering concepts. Continuing to acknowledge and support these difficulties using products like the LilyPad Arduino will hopefully continue to break down these barriers to agency and ludic engagement within the e-textiles domain. Whether personal or social experiences, combative or cooperative, the necessarily physical end products of e-textiles bring users' focus to their present existence in an embodied way that benefits CS, HCI, and crafting communities alike.

5.3.1 Designing and Making E-Textile Artifacts

Basic concepts related to electronics are easy to understand and packaged into beginner kits, such as the MaKey MaKey [177] (section 5.1.1) and LilyPad Arduino [182] (section 5.1.1) current flows through conductive materials in one direction, and this circuit must be connected in order for the current to flow. However, to do more than immediately react to a simple discretely connected circuit, the skills required by the designer and their learning curve spikes dramatically into more difficult territory. As I was primarily the crafter in the Fidget Widgets project above, and my previous experience in *Threadsteading* was mostly software hacking, I had to quickly learn the affordances of electrical components and how they could be incorporated with textiles.

Following Tutorials

Tutorials are in the same instructional group as patterns, but are generally used for masculine disciplines or for instructions that are meant to teach an overall skill. The same principles of a well-crafted pattern apply to a well-crafted tutorial in terms of increasing the audience's agency via clear instructions (link to other section). In the earliest brainstorming phases of the Fidget Widgets project, the authors surveyed online tutorials for inspiration and guidance. If someone had built a tutorial for it, it was not only possible, but we could incorporate its core elements into our design as one possible fidget interaction or feedback mechanism.

Some websites like Instructables [18] are made for many types of crafting and maker communities, not just electronics or soft circuits. Instructables is also user-curated, so there will always be new content. Other websites that produce and/or sell e-textile materials, such as Arduino [204], SparkFun [73], and Kitronik [102], also provide free tutorials. The primary source for our Fidget Widget inspiration was *How to Get What You Want*, a soft circuit DIY website with documentation on various e-textile experiments, created and curated by the art collective KOBAKANT [145]. The most interesting examples at the boundaries of soft circuits are found on artists' personal websites, such as Lisa Stark [185], and those of the *How to Get What You Want*'s primary authors Hannah Perner-Wilson [143] and Mika Satomi [164], although many projects do not include explicit tutorials.

People who come to electronics from a background of crafting are used to following tutorials as if they were any other craft pattern. The same properties of a high-quality crafting pattern are found in e-textile (and most other) tutorials: full materials and tools lists, many clear photos of in-progress and finished work, clear and logical layout, as well as having been verified and/or completed by someone else. The best tutorials are those that also offer insights into the logic behind making certain choices, such as ensuring a strong connection in a sewn circuit by bending the pins of some components into small circles to securely sew through.

Electrical Engineering Challenges

An unavoidable component of electronic textiles are the physical electronics that accompany the physical crafts. Understanding electrical engineering concepts deeply enough in order to expand outside of explicitly following tutorials is a very rare skill, much rarer than crafters being able to author their own patterns (section 4.4.2). Simply put, electrical engineering is an entirely new domain on top of the crafter's traditional background with few shared skills. The electrical engineering and programming difficulties are reflected in the style of kits and tutorials available, as well as in common perceptions of the e-textile hobby. For example, in advertising and teaching my university-level physical computing course, students derisively described wanting to do more than "make an LED light up." Nearly all kits that include additional electrical components include an array of LEDs, as they are by far the cheapest and easiest means of displaying a signal. Kits that include other components, such as the Arduino Starter Kit, offer many tutorials that are simply isolated tests of those components to demonstrate how they work. Having a wealth of these simplistic tutorials, rather than a progression of difficulty and skill acquisition, is an easy way to bloat tutorial count numbers while disappointing users. The more unique and interesting tutorials discussed in the last section, as well as their education potential if the tutorial is authored well, can help push through these misconceptions of over-simplicity.

Even if the project is well-designed and the user has all their components, debugging circuits is often more difficult than fixing crafting mistakes. There are generally two types of errors when handling circuits: design/assembly of the circuit, and faulty materials. Players of *eBee* (section 5.1.4) were seen collaborating to debug their circuits even during a competitive game (an example of the design/assembly style of error) [33]. It benefitted both players to better understand how the user's execution of the game contributed to the failed circuit. Design/assembly errors arise from misunderstandings of current and circuitry, or in some cases simply incorrectly following a tutorial. Deciphering a circuit diagram is a skill in and of itself, and unfortunately many tutorials simply supply a circuit diagram as the perfectly rendered example of the final circuit design. Even a tutorial that shows a screenshot of the final physically implemented circuit is not enough: a nest of wires may be extremely difficult or impossible to reverse-engineer without a step-by-step break down of the assembly process.

A secondary and often insidious error are those where an electrical component is faulty in some way, which can even be caused by the designer. For example, there are two sets of eyes on the Fidget Widget Dragon because the first set burned out due to not correctly using a resistor with the LED. The burn-out did not occur until after the permanent polymer clay baking process. Instead of scrapping the whole head, we used a two-part epoxy putty to extend the sculpted design so that we had at least one pair of functional eyes (Figure 5.8).

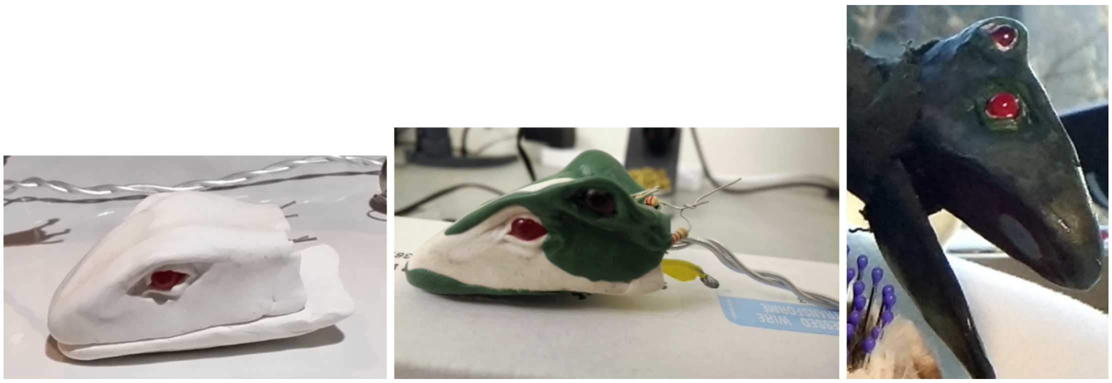


Figure 5.8: The grafting process of a second pair of eyes for the Dragon Fidget Widget toy. From left to right: the original head sculpted in clay, the unpainted addition of the two-part epoxy clay (green), and the final painted head that covers the mistake.

Finally, the reliability of certain types of electrical components, particularly self-monitoring sensors, are notoriously inaccurate. For example, GSR sensors used in *EmRoll* were noisy and often gave false data, which meant that when the game mechanics made use of that sensor's signals, gameplay broke down (section 5.1.3. Heart rate monitors are similarly inaccurate, especially those that use optical sensors on areas of the body that move a lot, such as arms, legs, and wrists. My experience with heart rate monitoring devices are that they give impossibly fluctuating signals on anything other than resting heart rates. Similarly, motion detectors such as IR sensors[11] must be pre-calibrated to the room and operate on a very limited range. Many of my students have been surprised by the limitations

[11]Infrared Radiation, or Infrared light, is outside the visible spectrum of light but is relatively easy to sense.

of these types of ready-made sensors. Part of what makes the MaKey MaKey work so well is that it works with very broad signal ranges and only needs to detect discrete activity. Clever use of sensors without relying on subtlety and fine accuracy leads to more successful projects.

Textile Challenges

Most textiles are inherently non-conductive, which is helpful for embedding secure circuits within and on them. However, conductive fabric, thread, and other fibres do not have any of the insulation that a coated wire has. What's more, many e-textile artists use the running stitch to sew conductive thread, a stitch that alternates being on top and bottom of a piece of fabric. This means that layering one piece of fabric on top of each other, each with isolated circuits, may cause the circuits to inadvertently touch. To avoid this issue, if the crafter has time, couching[12] the conductive thread with a non-conductive thread can protect it, although this process can be laborious depending on how well the conductive material needs to be covered. Alternative insulation methods include fabric glue, fabric paint, and fusible interfacing [73]. *eBee* uses a layered method to separate the positive and negative current connections on their LED island tiles, similar to two three-way road junctions on top of each other [33]. The upper path is laid upon a non-conductive piece of fabric that is carefully sewn over the lower path, enabling both paths to coexist on the same game piece without shorting each other.

Another design element of textiles to consider is their stretchability. Much of the clothing we wear and that get integrated into wearables are knit fabrics or

[12] An embroidery technique where the main thread (or other item) sits atop the fabric surface while a separate thread anchors it via intermittent stitches made over the item. Couching enables an item too big or cumbersome to âĂIJsewâĂİ through the fabric to be attached to the surface of fabric.

loose woven fabrics that contain different amounts of stretch. Gravity, exertion, or simply taking a wearable on and off can strain the electrical components to the point of breaking, or at least disconnecting. KOBAKANT marks their design sketches with "strain relief" whenever a design element is meant to ease these forces on connections to their components [144]. Methods for providing strain relief include adding anchoring stitches to the ends of pieces that should not move, such as USB cable, using purposefully circuitous sewing paths that have extra slack for stretching, and using stretch-friendly hand stitches such as the zig-zag or herringbone stitches.

5.3.2 Experiencing Computational Artifacts

It would be remiss to discuss the end product in the crafting processes without the experience of the end product (Rhodes' Four P "Press" that I folded into the product crafting process in section 1.2) [153]. The major benefit to e-textiles is that their final forms are generally familiar or evocative, and interacting with e-textiles as a much higher chance of agency and ludic engagement than while making e-textiles. The experiential qualities of e-textiles speak largely to the HCI community.

Being present in the moment and connected to the computational artifact are common themes among the Fidget Widgets and social wearables. Fidgeting is a behavior we all do, but our awareness of it is often negligible unless the object is meaningful. So much of what we understand as computational is digital — on phones or computers. Even multiplayer games are played entirely remotely and online or via a shared gaming system rather than directly with each other. Socially interdependent games enabled via physical wearables act as a means of connecting with other humans, even strangers, in ways we rarely do. E-textiles are a means of enabling these new experiences and integrating them into our daily lives.

Threadsteading* and *eBee

On surface description, the nature of *Threadsteading* [10] (section 4.3) and *eBee* [33] (section 5.1.4) are very similar: a center-starting, quilt-inspired hexagonal board game with hexagonal tiles and special island tiles that competitive players must claim in order to win. The two games do share a designer, and the hexagonal quilt aesthetic is appropriate for both games' relationship to quilts: *Threadsteading* being sewn on a quilt during gameplay and *eBee*'s game pieces being made of quilted hexes. Both games, as craft-infused board games, successfully lured and intrigued both technologically savvy audiences as well as crafters, across different ethnicities, genders, and age groups. However, each game's relationship to their technology, and thus the player's relationship to that technology, is very different.

Threadsteading requires a specific type of sewing machine and additional hardware to play, as well as general sewing machine expertise to setup and run the game. The player's actions on *Threadsteading* are separate from the game board — they act remotely via the control panel – but their actions are made permanent on the game board for them to take home. The sewing machine and various additional electronics make the setup very difficult to emulate and also intimidating or simply confusing to all types of audiences at first glance. The actual game board is only approximately 4.5 inches squared, embedded in a hoop and obscured by the sewing head, so there is a lot of additional clutter that distracts from the game itself. A user's initial feelings of physical agency are low, but quickly rise as they parse the presentation and logic of the game.

eBee's game board is not only lovingly hand-crafted, but is much more easily recognizable as a game board, and thus a game, than *Threadsteading*. Players tactically handle their game pieces and can examine the conductive materials very closely. The electrical components are not only much simpler, but they are core to the game's mechanics. The player approaches *eBee* with higher agency than *Threadsteading*. While the player may not walk away from *eBee* with a crafted artifact, they most likely will walk away having learned something new about circuits, conductive materials, and e-textiles as a whole. The experiences of *Threadsteading* and *eBee* are not necessarily better than the other, but they both leverage ludic engagement as a means of bridging the gaps between feminine crafting communities and masculine areas of technology.

E-textiles are an incredibly powerful educational platform for electronics and an innovative application of traditional textile art skills. However, the difficulty of circuit design and programming custom microcontrollers creates a huge barrier to entry for designing or adapting custom e-textile projects. The beginning feelings of agency of most outsiders examining electrical engineering is extremely low. Luckily, the incredible amount of resources online, the growing publications for electronics and e-textiles for all ages, and the creation of physical maker spaces have done much to enable and encourage designers and crafters to make e-textiles. The projects and artifacts discussed in this chapter show a sampling of different design complexities, design goals, and lessons involved in the creation and experience of e-textiles. It is my hope that treating electrical engineering and microcontroller programming as yet another crafting domain, and e-textiles as a multifaceted crafting domain, that feminine perceptions of the craft become more positive and inclusive.

Chapter 6

Conclusion

This dissertation has explored several projects across the three primary crafting processes of design, manufacture, and product. In these processes, there have been instances of high and low digital and physical agency, as well as many instances of ludic engagement. Computer science has contributed much research toward creativity support tools, which was the primary focus of chapter 3, as well as computational creativity, generative design, and mixed-initiative co-creativity that are themes throughout the whole dissertation. HCI has also contributed much to the user-centric design philosophy and practice of the research projects presented here. Finally, the domain of textile crafts is a rare one for computer science research, but computational crafts in general are a budding area of research. The communities of feminine crafts and masculine technology have demonstrably mixed, especially during experimental games like *Threadsteading* and e-textiles projects. This dissertation has presented an overview of budding interdisciplinary research that applies lessons of creativity, agency, and engagement across these different communities of academics and non-academics.

6.1 Research Contributions

As a precursor to the research questions, how have the revised crafting model, composed of crafting processes across design, manufacture, and product, supported this dissertation as an organizational framework for craft-related research? Figure 6.1 repeats Figure 1.3 from the introduction for ease of recollection.

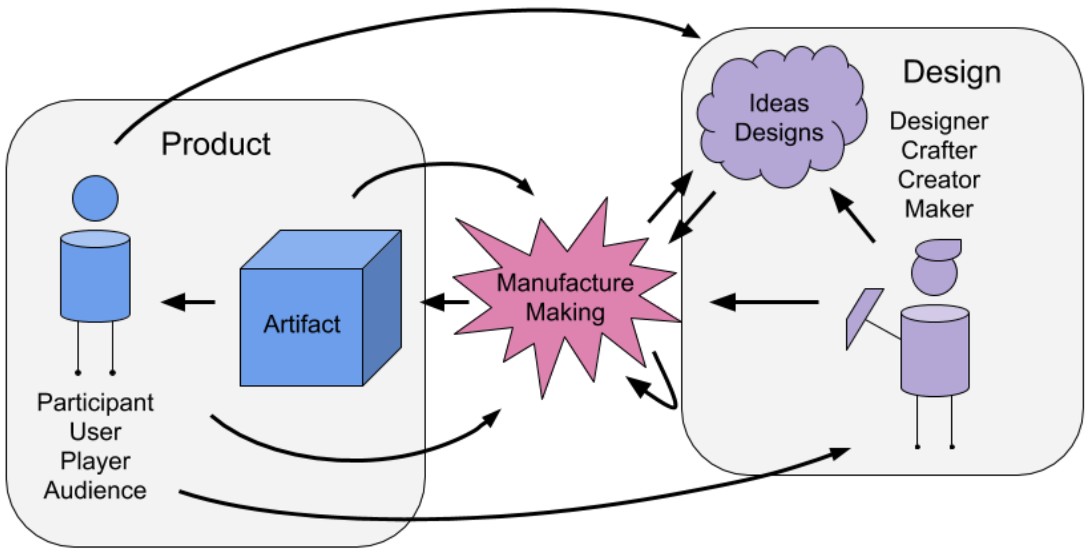


Figure 6.1: The revised crafting processes model as described in the introduction.

6.1.1 The Crafting Model Revisited

I condense and rename parts of Rhodes' Four P's model of creativity, made of person, process, product, and press, to better reflect the process-oriented view of crafting [153]. Design and planning are acknowledged as first-order creative tasks partially due to the focus of creativity support software in computer science, and because they are core tasks in crafting. Due to the nature of these nested processes, the creation of patterns is a crafting domain that follows the whole of the crafting processes model in and of itself. As a small note, the term "process" here is also extremely overloaded, so I rename it to manufacture. Finally, the influence of HCI as a user-experienced-focused discipline encouraged me to not only tie the "press" to the product, as a necessary lense of examining craft products, but also distribute its influence throughout the entire model. A crafter is almost always considering their audience throughout all stages in their crafting processes, even if their audience is themselves. The boundaries of this model are fuzzy in a realistic and necessary way. However, this framework is employed throughout this dissertation as an example of how it can be used to identify different crafting processes and their different agency and ludic engagement profiles. A single user's skill and comfort level, indicative of their agency and ludic engagement, varies depending on their creative task: from designing and adapting patterns, to the manual dexterity of manufacturing, and the interactive experiencing computational crafts.

6.1.2 Research Questions Revisited

How well have I addressed the research questions that provide the backbone to this document?

How can computationally-mediated interactions with physical and digital representations of traditional textile crafts affect the physical agency, digital agency, and ludic engagement of designers, manufacturers, and experiencers of textile crafts?

In order to address the disparate ways this takes place in different parts of the crafting process, this research question was addressed by breaking it down into three sub-questions - design, manufacture, and product - discussed below.

Mixed physical and digital interfaces that occur in co-operative manufacturing processes require a balance of mastery over physical machines and a willingness to adapt to their design affordances and constraints.

How can *digital* representations of textile crafts affect *designers'* agency and ludic engagement?

Interacting with digital representations to design patterns is informed by creativity support tools and computational creativity, involves much of HCI's user interface and experience design, and also necessitates careful abstraction of a specific textile craft subdomain. The two projects of primary focus in this chapter, the blackwork embroidery grammar and knitting machine compiler, present different domains and different approaches to representing their domain specific languages. The two projects show one example of a trade-off between control and surprise, which are key components of agency and ludic engagement.

The blackwork embroidery project specifically tackles a very restricted domain in order to make the design tool small and focused, where even a child playfully randomly pressing buttons could make an evocative design. The tool's reliance on a grammar with randomness in rule selection aimed for an experience of ludic engagement through surprise and playfulness. Experienced crafters familiar with embroidery design or software (high expected affordances) had a lower sense of digital agency due to the nature of a simple editor (low perceived affordances) and an unexpectedly restricted domain (low presented affordances). However, most users were novices with low expected affordances; they were not as bothered by the low number of presented and perceived affordances, and low agency did not hamper their reported experiences. The blackwork embroidery interface was not just simple with few presented affordances, but it had inexact controls: the few presented affordances had random or apparently-random outcomes. For crafters that expected *fine control* as part of the presented affordances, their agency was also lower due to this mismatch between their presented and perceived affordances.

For all of these cases of lower digital agency, their agency obstructed the freedom and pleasure of ludic engagement. Those users that embraced the limited and uncontrolled presented affordances experienced much more ludic engagement. The ease of the tool's use and its direct pipeline to hand and machine embroidery formats dramatically shortens the time invested during design iterations and possible future manufacturing processes. The user does not need any knowledge of embroidery to make use of the design tool, which accurately targets the domestic amateur textile crafter. The design space is functionally infinite while the design tool provides instant iteration tools, which enable the user to effectively execute a random walk artificial intelligence algorithm for as long as they want.

The knitting machine compiler project contrasts with the blackwork embroidery project in that it tackles a broader domain and formally models the machine operations within that domain. The domain of machine knitting is not as expansive as hand knitting, but the machine has a much higher propensity for catastrophic error. The compiler's primary purpose is abstracting away annoying, difficult, or troublesome details that are specific to the machine so that the user feels more confidence in their design decisions, which is an increase in agency. Adding additional design features to encourage playful and unexpected design choices for ludic engagement was not a priority for this project. Instead, the added layers of abstraction in this project allow the user to be more efficient and direct with their pattern design process. Arguably the most important part of the compiler is to prevent user error when making their designs, giving the user more confidence while allowing them to focus on the creatively productive activities using the knitting machine. While the compiler should not introduce serendipitous surprises into the design and still requires a fair bit of knitting machine knowledge, it does offer the user a much stronger sense of control through the confidence it provides, which ties directly into increased agency. However, the domain is very likely not as amateur-friendly as others in this dissertation, and thus agency of less skilled crafters is generally lower.

How can computationally-mediated textile craft *manufacture* affect crafters' agency and ludic engagement?

The use of tools while crafting is a common practice, but few would credit those tools with shared authorship[1] over the crafted product. Crafting tools, such as an older mechanical sewing machine, crochet hook, or embroidery frame, are passive entities that are used in order to carry out a function by the crafter. The computationally-mediated textile manufacture alludes to a computational element that is an active participant in the crafting process.

My project that is the focus of this chapter is *Threadsteading*: the two-player competitive territory control game played on a quilting machine and embroidery machine. The sewing machine bed is the game board, and the sewing panel is the game controller. The custom hardware attached to the machine enforces the game rules, keeps track of the game state, and calculates the score at the end. There are no pieces on the game board either; the entire game is played through the interface of the machine. At the end of play (at the end of the crafting manufacturing process), an embroidered piece of cloth is the textile product. The game *may* be played via a simulation, and the game progress may be sewn by hand, but that's not the experience that the designers nor the players want to have. The players choose to interface with the embroidery machine because playful computationally-mediated interaction with the physical embroidery process is the goal.

This game and others discussed in the chapter use computation as a means of enforcing or supporting game mechanics, or otherwise providing a context within which the crafting happens. The player may share their agency with the ma-

[1] "A work of shared authorship is one in which the player and the system collaboratively create a narrative artifact, ideally one which neither would have been capable of producing on their own" as being applied to crafting artifacts [162].

chine in making decisions. The player may follow the computer's instructions, surrendering authorial control. In these computationally-mediated manufacturing circumstances, the participants playfully engage with the computer and share agency, which leads to unknown patterns and/or unknown elements of the final craft product. Some crafters crave these refreshing means of making textile products, even when the feasibility or 'correctness' of the output cannot be assured.

How can the integration of computation with physical textile craft *products* affect the agency and engagement of those who craft and experience textile crafts?

The integration of computation and textile crafts is a subdomain of electronics in general: electric textiles. Contemporary crafting kits for these devices range from incredibly simple plug-and-play devices to far more complex microcontrollers that require some amount of programming skills, electrical engineering skills, and textile crafting skills. Because of this difficulty, I address both the challenges of making and the effects of using e-textiles as contributions toward agency and engagement.

My project that is the focus of this chapter is the fidget widgets: two toys that were designed with a wide range of crafting techniques and materials in order to entice and track the fidget behaviors of children who play with them. As a crafter that started with no experience with e-textiles, my partner on this project had to guide me on the affordances and constraints of how conductive and non-conductive materials would interact. The crafter's potential sense of physical agency is much lower in unknown domains compared to other craft domains. Even fundamental questions related to flow (section 2.1.4 such as where the user is compared to where they need to go, leave the user feeling lost with very low agency and no sense of ludic engagement. However, a crafter's tenacity to learn is dependent on the individual, for the same reasons only some crafters want the challenge of adapting patterns 4.4.2. The potential of making something for the explicit purpose of being played with also encourages a sense of ludic engagement while crafting. The lessons learned via Perspectives 1 and 2 in pattern design and pattern interpretation come to a head in Perspective 3, where nearly all crafters coming from the space of domestic amateur textile crafts share a similar novice skill level in electrical engineering.

Once complete, the audience perspective with just the crafted product is entirely different. Children used the fidget toys as inspiration for designing their own, as well as discovering what fidget activities they found most engaging. The experience of adults using e-textiles, both the fidget toys and other e-textile projects, were similarly engaged. All participants necessarily explored the e-textiles with all their available senses, particularly those of touch, which in a computing context and textile exhibition space is very rare. The participant may not feel comfortable performing all actions they normally would, such as destroying the artifacts, which conflicts with agency. However, most of the the participant's desires (feelings of agency) are met with knowledge of agency as they are able to touch and participate with the artifacts. Wishes for further interactions with the artifacts are often met with positive reinforcement, which increases their sense of agency with these artifacts. These products were also able to take advantage of the physicality of craft products and the familiarity with textiles to invite ludic engagement. Partly because these materials are fairly recent developments, and partly due to the skill level required, these kinds of experiences are rare compared to traditional crafts. Their current novelty adds to the wonder and appreciation of the textile craft product.

Now I return to the overall research question.

How can computationally-mediated interactions with physical and digital representations of traditional textile crafts affect the physical agency, digital agency, and ludic engagement of designers, manufacturers, and experiencers of textile crafts?

The most overarching concepts of computationally-mediated interaction with traditional textile crafts are the challenges of design (of patterns and of the final crafts), crafter skill level and comfort, and the experience and perception of the crafted product.

Digital representations provide a safer and more efficient means of ideation without consuming material resources, needing tools, or securing the space necessary for design. Ideally, digital design spaces help abstract away the difficult, confusing, or tedious tasks related to this crafting process. Efficiency, ease of use, and successful user interaction design offer optimal design agency while possibly encouraging serendipitous ludic engagement.

Physical representations require respect for the textile craft affordances, but computationally-mediated physical crafts offer an entirely different experience due to these new computational perspectives. A crafter that allows a machine and/or a piece of software to provide input to their crafting process is sharing their agency: accepting unknown and/or unexpected possibilities. Without having full design control, the crafter creates for the pleasure of doing so, for pure ludic engagement.

Finally, crafters and audiences of computationally-mediated physical textile crafts work with physical agency and physically interactive forms of ludic engagement. The physical affordances and constraints of electronics along with textiles drastically increases the skill required in order to master complex designs, but drastically increases the kinds of designs a crafter can make. This expanded design space of two domains combined allows for both deeper frustrations and less agency, but also a broad range of interactive designs that increase agency. Due to designing directly for active physical interaction with a curious audience, crafters often feel an increase in ludic engagement as they imagine and playtest these interactive experiences. Without having to deal with the difficulty of crafting skill in two domains, audiences of computationally-mediated physical textile crafts experience a much cleaner increase in physical agency and ludic engagement. The audience necessarily experiences the artifacts in the present moment, which engages feelings of agency and invites curious exploration of the likely unfamiliar artifact. Multiplayer experiences, games in particular, often involve touching or directly interacting with someone else along with the e-textile product, explicitly empowering the knowledge of physical agency. The feedback given by these e-textiles, designed by textile crafters, also offer new forms of ludic engagement.

This dissertation presents an overview of a nested and overlapping set of processes, the crafting processes. Domestic amateur textile crafters as the target domain offer a subset of crafting activities that both have a long history of software and hardware support, and yet have regularly published pioneering research. How contemporary software and crafting machines interact with the domestic amateur textile crafter drastically differ between different stages of the crafting processes. The crafter has different specific goals and accomplishes different activities during each process, and thus the means for easing the difficulties of those processes and offering more informed choices increases the crafter's overall agency. The manufacturing process and end products are arguably the most evocative when they are interactive, especially with machines as a new potential crafting partner. The crafter and audience engage in playful curiosity through the mix of familiar and foreign design elements, which is an increase in ludic engagement. Overall, both of the domains of computation and textile crafts are enriched with the inclusion of the other.

6.2 Future Work

Many crafters of a domain can follow a pattern, or learn the skills in order to do so. These skills are well-documented and can be self-taught due to the wealth of free information on the internet. However, making designs continues to be a task that many crafters do not do. Software design tools are meant to enable users to make these patterns, but a majority of crafters still do not use them. The following future work aims to use the lessons of agency and ludic engagement to understand this problem, as well as apply the lessons in order to create a design tool that increases agency and ludic engagement.

The range of current design tools has a heavy bias toward productivity, especially those for craft design, as we've seen in machine embroidery software and quilt design. Casual creator tools are breaking out of this mold, but they almost never sync up with physical realization of crafts and are usually shallow in their representation of a domain.

If a user is well-versed in productivity-focused software, or if the software user interface is designed for multi-level navigation that aims not to overwhelm novice (or amateur) users, a user may have a high sense of agency. The user should know the craft domain and have an idea of what to make (expected affordances) and be able to actuate their design ideas via finding and using (percieved affordances) the appropriate tools for their task (presented affordances). However, if a user does not have a clear idea of what to make in a situation with ample percieved affordances, their agency drops, as nearly all tools passively follow the user's direction only exactly as specified. An amateur that does not know the breadth of their crafting domain's design space will be easily overwhelmed by the percieved affordances and not experience high agency nor ludic engagement. This low agency is what leads many people to lack the confidence to use design software and/or make patterns at all. Productivity-focused software also prioritizes task definition and completion, as well as efficiency. Easy and playful exploration of the domain's design space for inspiration, imitation, or improvisation is not efficient, nor is it a well-defined task. Even if an amateur does manage to have high agency in interacting with the tool and knows what they want to make, they likely will not experience ludic engagement while making their design.

Research has begun on a digital design tool: an expansion of the blackwork embroidery project that encompases general machine (and hand) embroidery. However, unlike the existing productivity-focused embroidery design tools, this research prioritizes ludic engagement. Inspired by projects such as Kid-Pix [82], the design tool will push discovery, play, and serendipity over fine embroidery stitch control, while still allowing fine control. The software will not only passively attempt to correct for the machine embroidery errors shown in Figure 4.21, but actively engage the user in design suggestions and support when using the embroidery machine for its final sew-out. The enhanced artificial intelligence that offers proactive help will lower the type of high agency that belongs to the mastery of existing productivity tools. However, the additional support will further buffer novices against failure and further abstract away the complexities of machine embroidery, which will increase the sense of agency for amateurs. This research primarily targets the design and manufacture chapters of this dissertation and will hopefully help alleviate the fear of machine embroidery pattern design by amateurs.

6.2.1 Future Motivations

Interdisciplinary work is both one of the hardest paths through academia and one of the most important. While research communities necessarily have boundaries as part of their selection criteria for conference contributions, interdisciplinary work that only partially satisfy these criteria are often excluded. The mixture of technical research projects and non-technical research questions in this dissertation are evidence of what was required to make it past academic community gatekeeping. For example, the blackwork embroidery project was repeatedly rejected, even from procedural content generation communities, as not generating the right kind of content for the right kind of purposes, until it found its home as a general expression of computational creativity. Applied computing should be a shared foundational aspect of most (if not all) research communities as a catch-all for instances of practical applications of academic research. Accepting wide applications of community research topics to new domains should be encouraged rather than spurned.

Respecting applied computing as a category of meaningful research contributions is one means of being more inclusive of interdisciplinary work. A more radical suggestion of increased inclusivity is to consider related but non-academic communities as being respected contributors (not just case studies), both in their people and their areas of expertise. In the case of this dissertation, domestic amatuer crafters are not only my audience, but also my participants, research collaborators, and, ideally, my audience: the audience that will actually apply and use this work rather than see it as a research novelty (or, to some, an aberration). However, most amateur crafters will never read an academic research paper. The incompatibility between scientific[2] academic research presentation and most of

[2]The humanities are slightly better at audience outreach.

the human population speaks to our communities' privilege and exclusivity.

The long-standing traditions of exclusive language and privileged topics in academia are barriers I intend to continue eroding. Diversity in the "workplace" of our academic communities can easily reap the same benefits as those found in other workplace contexts, including things like retention, creativity, innovation, and engagement [174, 122]. Our research should be made legible to communities outside academia so that our research is practically used in contexts outside of controlled case studies[3]. Every one of my research projects has been from a position of implicit community outreach: to designers, crafters, and gamers. I will continue to make my research more accessible to communities outside of academia, to speak to my primary audience of crafters. Improving not just their user experience but their knowledge and experience in their crafting domains, for their personal advancement, is my primary motivation.

[3] Not to say that no research has ever been used outside of its academic contexts, but it should be the norm rather than the anomaly.

Bibliography

[1] Ravelry forum discussion board: lazy, stupid, godless. https://www.ravelry.com/discuss/lazy-stupid-and-godless/3718985/1-25. Accessed 21-September-2019.

[2] Embroidermodder. http://embroidermodder.org/news0.html, 2013. Accessed 21-September-2019.

[3] Adafruit. Conductive fiber. https://www.adafruit.com/product/1088. Accessed 21-September-2019.

[4] Adafruit. Knit conductive fabric. https://www.adafruit.com/product/1364. Accessed 21-September-2019.

[5] Adafruit. Stainless thin conductive thread. https://www.adafruit.com/product/640. Accessed 21-September-2019.

[6] Adafruit. Woven conductive fabric. https://www.adafruit.com/product/1168. Accessed 21-September-2019.

[7] Adafruit. Adafruit industries. https://www.adafruit.com, 2005. Accessed 21-September-2019.

[8] Adafruit Blog. Celebrating 10 years of the lilypad arduino. https://blog.adafruit.com/2017/10/11/celebrating-10-years-of-the-lilypad-arduino/. Accessed 21-September-2019.

[9] Glenn Adamson. *Thinking through craft*. 2007.

[10] Lea Albaugh, April Grow, Chenxi Liu, James McCann, Gillian Smith, and Jennifer Mankoff. Threadsteading: Playful interaction for textile fabrication devices. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 285–288. ACM, 2016.

[11] all yarns are beautiful. Ayab - all yarns are beautiful. http://ayab-knitting.com/index.html#features, 2014. Accessed 21-September-2019.

[12] Miguel Bruns Alonso. *Relax!: Inherent Feedback During Product Interaction to Reduce Stress*. Bruns Alanso, 2010.

[13] Amazon.com, Inc. Amazon.com. https://www.amazon.com, 1996. Accessed 21-September-2019.

[14] Amazon.com, Inc. Handmade by amazon. https://services.amazon.com/handmade.html, 2015. Accessed 21-September-2019.

[15] Annchen. What do you look for in patterns? Online Forum Message Posted to https://www.craftster.org/forum/index.php?topic=452293.msg5482902#msg5482902. Accessed 21-September-2019.

[16] Kate Atherley. On untidy row repeats. https://www.stitchmastery.com/on-untidy-row-repeats-guest-blog-from-kate-atherley/. Accessed 21-September-2019.

[17] Autodesk. Autocad, 1982.

[18] Autodesk Inc. Instructables. https://www.instructables.com, 2005. Accessed 21-September-2019.

[19] Nick Babich. The 4 golden rules of ui design. https://theblog.adobe.com/4-golden-rules-ui-design/. Accessed 21-September-2019.

[20] Elizabeth Barber. A tradition with a reason. In Jessica Hemmings, editor, *The textile reader*, page 322. Berg Oxford, 2012.

[21] Sean Baron. Cognitive flow: The psychology of great game design. https://www.gamasutra.com/view/feature/166972/cognitive_flow_the_psychology_of_.php. Accessed 21-September-2019.

[22] Giovanna Best. 15 things i hate most about crafting. https://www.theodysseyonline.com/true-life-sorority-girl-that-hates-crafting. Accessed 21-September-2019.

[23] Debarun Bhattacharjya. Preference models for creative artifacts and systems. In *Proc. Int. Conf. Comput. Creativity*, pages 52–59, 2016.

[24] Adda Birnir. Lilypad arduino. https://www.huffpost.com/entry/lilypad-arduino_b_1507486, 2012. Accessed 21-September-2019.

[25] BitchinStitchesByLiz. Cross stitch pattern pdf - funny subversive consent and dissent. https://www.etsy.com/listing/658937086/cross-stitch-pattern-pdf-funny. Accessed 21-September-2019.

[26] Elisabeth Augusta Borchgrevink. Elisabeth augusta borchgrevink's flickr photostream. `https://www.flickr.com/photos/elisabethaugusta/`. Accessed 21-September-2019.

[27] Brenda Brathwaite and Ian Schreiber. Challenges for game designers, charles river media. *Inc., Rockland, MA*, 2008.

[28] Leah Buechley. A construction kit for electronic textiles. In *2006 10th IEEE international symposium on wearable computers*, pages 83–90. IEEE, 2006.

[29] Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. The lilypad arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 423–432. ACM, 2008.

[30] Leah Buechley, Nwanua Elumeze, and Michael Eisenberg. Electronic/computational textiles and children's crafts. In *Proceedings of the 2006 conference on Interaction design and children*, pages 49–56. ACM, 2006.

[31] Gülen Çağdaş. A shape grammar: the language of traditional turkish houses. *Environment and Planning B: Planning and Design*, 23(4):443–464, 1996.

[32] Jaime R Carbonell. Mixed-initiative man-computer instructional dialogues. Technical report, Bolt Beranek & Newman Inc Cambridge MA, 1970.

[33] Isabella Carlsson, Jeanie Choi, Celia Pearce, and Gillian Smith. Designing ebee: a reflection on quilt-based game design. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, page 24. ACM, 2017.

[34] Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A Otaduy. Yarn-level simulation of woven cloth. *ACM Transactions on Graphics (TOG)*, 33(6):207, 2014.

[35] Gabriel Cirio, Jorge Lopez-Moreno, and Miguel A Otaduy. Efficient simulation of knitted cloth using persistent contacts. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 55–61. ACM, 2015.

[36] John Cloud. Better learning through fidgeting, 2009.

[37] Simon Colton. Creativity versus the perception of creativity in computational systems. In *AAAI spring symposium: creative intelligent systems*, volume 8, 2008.

[38] Simon Colton, John William Charnley, and Alison Pease. Computational creativity theory: The face and idea descriptive models. In *ICCC*, pages 90–95, 2011.

[39] The Electric Quilt Company. About electric quilt. `https://electricquilt.com/home/about-electric-quilt/`. Accessed 21-September-2019.

[40] Kate Compton, Joseph C Osborn, and Michael Mateas. Generative methods. In *The Fourth Procedural Content Generation in Games workshop, PCG*, volume 1, 2013.

[41] Katherine E Compton. *Casual Creators: Defining a Genre of Autotelic Creativity Support Systems*. PhD thesis, UC Santa Cruz, 2019.

[42] Edutech Wiki Contributors. Embroidery format dst - edutech wiki. `https://edutechwiki.unige.ch/en/Embroidery_format_DST`, 2019. Accessed 21-September-2019.

[43] Robert L Cook, Loren Carpenter, and Edwin Catmull. The reyes image rendering architecture. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 95–102. ACM, 1987.

[44] Sarah Corbett. *How to be a Craftivist*. Unbound Publishing, 2017.

[45] Peter Cottrell, April Grow, and Katherine Isbister. Soft-bodied fidget toys: A materials exploration. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 42–48. ACM, 2018.

[46] National Research Council et al. *Beyond productivity: Information technology, innovation, and creativity*. National Academies Press, 2003.

[47] Joel Couture. Alt.ctrl.gdc showcase: Threadsteading. `https://www.gamasutra.com/view/news/266886/AltCtrlGDC_Showcase_Threadsteading.php`, 2016. Accessed 21-September-2019.

[48] Craftaholique. How to make brick stitch star earrings. `https://craftaholique.com/jewellery-making/how-to-make-brick-stitch-star-earrings/`. Accessed 21-September-2019.

[49] Craftster. Craftster. `https://www.craftster.org`, 2000. Accessed 21-September-2019.

[50] Matthew Crawford. *The Case for Working with Your Hands: Or why office work is bad for us and fixing things feels good*. Penguin UK, 2010.

[51] crazygirlfriend. I hate crafting, and you probably do too. https://totalsororitymove.com/i-hate-crafting-and-you-probably-do-too/. Accessed 21-September-2019.

[52] Mihaly Csıkszentmihályi. Beyond boredom and anxiety, 1975.

[53] Mihaly Csikszentmihalyi. Flow and the psychology of discovery and invention. *HarperPerennial, New York*, 39, 1997.

[54] Linda D. 20 hilarious pinterest fails. https://www.boredpanda.com/funny-pinterest-fails/. Accessed 21-September-2019.

[55] Suzanne B da Câmara, Rakshit Agrawal, and Katherine Isbister. Identifying children's fidget object preferences: Toward exploring the impacts of fidgeting and fidget-friendly tangibles. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pages 301–311. ACM, 2018.

[56] Ella Dagan, Elena Márquez Segura, Ferran Altarriba Bertran, Miguel Flores, and Katherine Isbister. Designing'true colors': A social wearable that affords vulnerability. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 33. ACM, 2019.

[57] Ella Dagan, Elena Márquez Segura, Ferran Altarriba Bertran, Miguel Flores, Robb Mitchell, and Katherine Isbister. Design framework for social wearables. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, DIS '19, pages 1001–1015, New York, NY, USA, 2019. ACM.

[58] Oyvind Nydal Dahl. *Electronics for Kids: Play with Simple Circuits and Experiment with Electricity!* No Starch Press, 2016.

[59] Department for Digital, Culture, Media, & Sport. Libraries and makerspaces. https://www.gov.uk/government/publications/libraries-and-makerspaces/libraries-and-makerspaces. Accessed 21-September-2019.

[60] Digital Design and Innovation. Threadsteading! interview with: April grow from disney reaserch. https://www.youtube.com/watch?v=oXPhdS4SD-w&t=4s, 2016. Accessed 21-September-2019.

[61] dharaniw. I will do knit programming by using shima seiki sds one. https://www.fiverr.com/dharaniw/do-knit-programming-by-using-shima-seiki-sds-one-apex3. Accessed 21-September-2019.

[62] Camrbidge Dictionary. Craft. https://dictionary.cambridge.org/dictionary/english/craft. Accessed 21-September-2019.

[63] Merriam-Webster Dictionary. Craft. https://www.merriam-webster.com/dictionary/craft. Accessed 21-September-2019.

[64] Oxford Dictionary. Craft. https://www.lexico.com/en/definition/craft. Accessed 21-September-2019.

[65] Peter Dormer. *The culture of craft*. Manchester University Press, 1997.

[66] Etsy, Inc. Etsy. https://www.etsy.com, 2005. Accessed 21-September-2019.

[67] FeltCrafts, LLC. The new feltcrafts table-top needle felting machine. http://feltcrafts.com/new-machines.html. Accessed 21-September-2019.

[68] Chris Franses. Eletric quilt (eq 8). https://ukqu.co.uk/electric-quilt-eq-8/. Accessed 21-September-2019.

[69] Free Epic Cross Stitch Sewing Jigsaw Puzzle games. Cross stitch quest - sewing pattern mania (2.27.0). https://play.google.com/store/apps/details?id=air.com.crossstitchquest, 2018. Mobile Application. Accessed: 2019-09-21.

[70] Game Developer's Conference. Here's the game lineup for gdc 2016's alt.ctrl.gdc showcase. https://www.gdconf.com/news/heres_the_game_lineup_for_gdc_/. Accessed 21-September-2019.

[71] David Gauntlett. *Making is connecting*. John Wiley & Sons, 2013.

[72] Elizabeth Geddes and Moyra McNeill. *Blackwork embroidery*. Courier Corporation, 1976.

[73] Gella. Insulation techniques for e-textiles. https://learn.sparkfun.com/tutorials/insulation-techniques-for-e-textiles/all. Accessed 21-September-2019.

[74] Amy Gibson. Sampler layout tutorial & an eq7 giveaway! http://www.stitcherydickorydock.com/sampler-layout-tutorial-an-eq7-giveaway/. Accessed 21-September-2019.

[75] Google. Google glass. https://www.google.com/glass/start/. Accessed 21-September-2019.

[76] Google. Project jacquard. https://atap.google.com/#project-jacquard. Accessed 21-September-2019.

[77] April Grow, Melanie Dickinson, Johnathan Pagnutti, Noah Wardrip-Fruin, and Michael Mateas. Crafting in games. *Digital Humanities Quarterly*, 11(4), 2017.

[78] April Grow, Michael Mateas, and Noah Wardrip-Fruin. Blackwork embroidery pattern generation using a parametric shape grammar. In *Proceedings of the 8th International Conference on Computational Creativity*, 2017.

[79] V. Guljajeva and M. Canet. Knitic open hardware knitting machine. http://www.knitic.com, 2012. Accessed 21-September-2019.

[80] Fiona Hackney. Quiet activism and the new amateur: The power of home and hobby crafts. *Design and Culture*, 5(2):169–193, 2013.

[81] Kate Haxell and Becky Hogg. *The Stitch Bible*. Capricorn Link Aust Pty Limited, 2012.

[82] Craig Hickman. Kid pix, 1989.

[83] Becky Hogg. Blackwork: Royal school of needlework essential stitch guides, 2010.

[84] Joshua Holden. The graph theory of blackwork embroidery. 2007.

[85] Shih-Wen Hsiao and Ching-Hai Chen. A semantic and shape grammar based approach for product design. *Design studies*, 18(3):275–296, 1997.

[86] Scott E Hudson. Printing teddy bears: a technique for 3d printing of soft interactive objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 459–468. ACM, 2014.

[87] Yuki Igarashi, Takeo Igarashi, and Hiromasa Suzuki. Knitty: 3d modeling of knitted animals with a production assistant interface. In *Eurographics (Short Papers)*, pages 17–20. Citeseer, 2008.

[88] CLO Virtual Fashion Inc. Marvelous designer. http://marvelousdesigner.com, 2010. Accessed 21-September-2019.

[89] IndieCade. International festival of independent games. https://www.indiecade.com. Accessed 21-September-2019.

[90] infinitywaltz. Indiecade 2016 official award winners âĂŞ the complete list. https://indiegamereviewer.com/30886-2/, 2016. Accessed 21-September-2019.

[91] Katherine Isbister, Kaho Abe, and Michael Karlesky. Interdependent wearables (for play): A strong concept for design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 465–471. ACM, 2017.

[92] Anna Jordanous. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation*, 4(3):246–279, 2012.

[93] Judith. Module 2 chapter 2: A tonal column of stitches. `https://judithstitches.wordpress.com/2014/10/09/module-2-chapter-2/`. Accessed 21-September-2019.

[94] Yasmin B Kafai, Eunkyoung Lee, Kristin Searle, Deborah Fields, Eliot Kaplan, and Debora Lui. A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE)*, 14(1):1, 2014.

[95] Jonathan M Kaldor, Doug L James, and Steve Marschner. Simulating knitted cloth at the yarn level. In *ACM Transactions on Graphics (TOG)*, volume 27, page 65. ACM, 2008.

[96] Jonathan M Kaldor, Doug L James, and Steve Marschner. Efficient yarn-based cloth with adaptive contact linearization. In *ACM Transactions on Graphics (TOG)*, volume 29, page 105. ACM, 2010.

[97] Michael Karlesky and Katherine Isbister. Fidget widgets: secondary playful interactions in support of primary serious tasks. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 1149–1154. ACM, 2013.

[98] Michael Karlesky and Katherine Isbister. Designing for the physical margins of digital workspaces: fidget widgets in support of productivity and creativity. In *Proceedings of the 8th international conference on tangible, embedded and embodied interaction*, pages 13–20. ACM, 2014.

[99] Michael Karlesky and Katherine Isbister. Understanding fidget widgets: Exploring the design space of embodied self-regulation. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, page 38. ACM, 2016.

[100] Kathryn. Is there such a thing as a crochet machine? `https://www.crochetconcupiscence.com/2011/06/is-there-such-a-thing-as-a-crochet-machine/`. Accessed 21-September-2019.

[101] Joan L Kirsch and Russell A Kirsch. The structure of paintings: formal grammar and design. *Environment and Planning B: Planning and Design*, 13(2):163–176, 1986.

[102] Kitronik. E-textiles and wearables tutorials & resources. `https://www.kitronik.co.uk/blog/e-textiles-wearables-tutorials-resources/`. Accessed 21-September-2019.

[103] Stephen Knott. *Amateur craft: history and theory*. Bloomsbury Publishing, 2015.

[104] Hank Koning and Julie Eizenberg. The language of the prairie: Frank lloyd wright's prairie houses. *Environment and planning B: planning and design*, 8(3):295–323, 1981.

[105] Golden Krishna. *The best interface is no interface: The simple path to brilliant technology*. Pearson Education, 2015.

[106] Carolyn Lamb, Daniel G Brown, and Charles LA Clarke. Evaluating computational creativity: An interdisciplinary tutorial. *ACM Computing Surveys (CSUR)*, 51(2):28, 2018.

[107] Eunkyoung Lee, Yasmin B Kafai, Veena Vasudevan, and Richard Lee Davis. Playing in the arcade: Designing tangible interfaces with makey makey for scratch games. In *Playful user interfaces*, pages 277–292. Springer, 2014.

[108] Helen Leigh. *The Crafty Kids Guide to DIY Electronics: 20 Fun Projects for Makers, Crafters, and Everyone in Between*. McGraw-Hill Education TAB, 2019.

[109] Catherine Amoroso Leslie. *Needlework through history: An encyclopedia*. Greenwood Publishing Group, 2007.

[110] Faythe Levine and Cortney Heimerl. *Handmade nation*. 2009.

[111] Robert Lewicki. Knit mania (1.3). `https://play.google.com/store/apps/details?id=com.rainbowaferateam.knitmania`, 2019. Mobile Application. Accessed: 2019-09-21.

[112] Antonios Liapis, Gillian Smith, and Noor Shaker. Mixed-initiative content creation. In *Procedural content generation in games*, pages 195–214. Springer, 2016.

[113] Joseph Carl Robnett Licklider. Man-computer symbiosis. *IRE transactions on human factors in electronics*, (1):4–11, 1960.

[114] Yong-Jin Liu, Dong-Liang Zhang, and Matthew Ming-Fai Yuen. A survey on cad methods in 3d garment design. *Computers in Industry*, 61(6):576–593, 2010.

[115] Google LLC. Youtube, 2005.

[116] Soft Byte Ltd. Designaknit. https://www.softbyte.co.uk/designaknit.htm, 1999. Accessed 21-September-2019.

[117] Alexis C. Madrigal. Skyknit: How an ai took over an adult knitting community. https://www.theatlantic.com/technology/archive/2018/03/the-making-of-skyknit-an-ai-yarn/554894/, 2018. Accessed 21-September-2019.

[118] Make:. A game of stitches: Territory control through embroidery. https://www.youtube.com/watch?v=tgnitvfP4iM&t=9s, 2016. Accessed 21-September-2019.

[119] Make:Community. Makerspace directory. https://makerspaces.make.co. Accessed 21-September-2019.

[120] Joanna Mann. Towards a politics of whimsy: Yarn bombing the city. *Area*, 47(1):65–72, 2015.

[121] Alissa Marquess. 10 ways to guarantee you will hate doing kids crafts. https://bouncebackparenting.com/10-way-to-guarantee-you-will-hate-doing-kids-crafts/. Accessed 21-September-2019.

[122] Kristina Martic. Top 10 benefits of diversity in the workplace. https://www.talentlyft.com/en/blog/article/244/top-10-benefits-of-diversity-in-the-workplace-infographic-\included. Accessed 21-September-2019.

[123] Peter Mawhorter, Carmen Zegura, Alex Gray, Arnav Jhala, Michael Mateas, and Noah Wardrip-Fruin. Choice poetics by example. In *Arts*, volume 7, page 47. Multidisciplinary Digital Publishing Institute, 2018.

[124] James McCann, Lea Albaugh, Vidya Narayanan, April Grow, Wojciech Matusik, Jennifer Mankoff, and Jessica Hodgins. A compiler for 3d machine knitting. *ACM Transactions on Graphics (TOG)*, 35(4):49, 2016.

[125] Caty McCarthy. Threadsteading is a strategy game you play by sewing. https://killscreen.com/articles/threadsteading-is-a-strategy-game-you-play-by-sewing/, 2016. Accessed 21-September-2019.

[126] Micropro. Wordstar, 1979.

[127] Andrea Minuto and Anton Nijholt. Smart material interfaces as a methodology for interaction: a survey of smis' state of the art and development. In *Proceedings of the second international workshop on Smart material interfaces: another step to a material future*, pages 1–6. ACM, 2013.

[128] Victoria Mitchell. Textiles, text, and techne. In Jessica Hemmings, editor, *The textile reader*, page 322. Berg Oxford, 2012.

[129] Giovanni B Moneta and Mihaly Csikszentmihalyi. The effect of perceived challenges and skills on the quality of subjective experience. *Journal of personality*, 64(2):275–310, 1996.

[130] James W Moore. What is the sense of agency and why does it matter? *Frontiers in psychology*, 7:1272, 2016.

[131] Ann J Morrison, Peta Mitchell, and Margot Brereton. The lens of ludic engagement: evaluating participation in interactive art installations. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 509–512. ACM, 2007.

[132] Oli Mould. Tactical urbanism: The new vernacular of the creative city. *Geography Compass*, 8(8):529–539, 2014.

[133] Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. Interactive construction: interactive fabrication of functional mechanical devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 599–606. ACM, 2012.

[134] Jeanne Nakamura and Mihaly Csikszentmihalyi. Flow theory and research. *Handbook of positive psychology*, pages 195–206, 2009.

[135] Ramsey Nasser. Swordfight. `http://nas.sr/swordfight/`. Accessed 21-September-2019.

[136] Nerdigurumi. A word about requests for help. `https://nerdigurumi.com/2013/12/a-word-about-requests-for-help.html`, 2013. Accessed 21-September-2019.

[137] Dong Nguyen. Flappy bird. `https://flappybird.io/`, 2013. Mobile Application. Web edition Accessed: 2019-09-21.

[138] David G Novick and Stephen Sutton. What is mixed-initiative interaction. In *Proceedings of the AAAI spring symposium on computational models for mixed initiative interaction*, volume 2, page 12, 1997.

[139] Openknit. Openknit: open source digital knitting. http://www.openknit.org, 2014. Accessed 21-September-2019.

[140] Thomas Oppong. For a more creative brain, embrace constraints. https://www.inc.com/thomas-oppong/for-a-more-creative-brain-embrace-constraints.html. Accessed 21-September-2019.

[141] Despina Papadopoulos. Material implications: Hug jackets. https://pixelpeppy.com/hugjackets/. Accessed 21-September-2019.

[142] Paradise Fibres. Basics of 3d needle felting. https://paradisefibersblog.wordpress.com/2014/04/01/basics-of-3d-needle-felting/. Accessed 21-September-2019.

[143] Hannah Perner-Wilson. Plusea. https://www.plusea.at/. Accessed 21-September-2019.

[144] Hannah Perner-Wilson and Mika Satomi. Strain relief. https://www.kobakant.at/DIY/?p=7075. Accessed 21-September-2019.

[145] Hannah Perner-Wilson and Mika Satomi. How to get what you want. *KOBAKANT DIY Wearable Technology Documentation*, 2010.

[146] Eva Petersson. *Non-formal learning through ludic engagement within interactive environments*. Malmö högskola, Lärarutbildningen, 2006.

[147] Philips. Mind spheres. http://www.design.philips.com/about/design/designportfolio/design_futures/mind_spheres.page. Accessed July-2015.

[148] Pinterest, Inc. Pinterest. https://www.pinterest.com, 2009. Accessed 21-September-2019.

[149] Michael J Pugliese and Jonathan Cagan. Capturing a rebel: modeling the harley-davidson brand through a motorcycle shape grammar. *Research in Engineering Design*, 13(3):139–156, 2002.

[150] David Pye. *The nature and art of workmanship*. 1968.

[151] Jonathan Ragan-Kelley, Andrew Adams, Sylvain Paris, Marc Levoy, Saman Amarasinghe, and Frédo Durand. Decoupling algorithms from schedules for easy optimization of image processing pipelines. 2012.

[152] Ravelry, LLC. Ravelry. https://www.ravelry.com, 2007. Accessed 21-September-2019.

[153] Mel Rhodes. An analysis of creativity. *The Phi Delta Kappan*, 42(7):305–310, 1961.

[154] Gabriela T Richard and Yasmin B Kafai. Making physical and digital games with e-textiles: a workshop for youth making responsive wearable games and controllers. In *Proceedings of the 14th international conference on interaction design and children*, pages 399–402. ACM, 2015.

[155] Tania Richter. Yarn quest 2017 - heroes of yarnia rpj mkal. https://www.ravelry.com/patterns/library/yarn-quest-2017---heroes-of-yarnia-rpg-mkal, 2016. Accessed 21-September-2019.

[156] Tania Richter. Yarnia - the grand quest blanket mkal. https://www.ravelry.com/patterns/library/yarnia---the-grand-quest-blanket-mkal, 2017. Accessed 21-September-2019.

[157] Tania Richter. Storied yarns - a yarn quest knitting rpg. https://www.kickstarter.com/projects/taniarichter/welcome-to-yarnia-a-yarn-quest-knitting-rpg, 2018. Accessed 21-September-2019.

[158] Yvonne Rogers, Jeni Paay, Margot Brereton, Kate L Vaisutis, Gary Marsden, and Frank Vetere. Never too old: engaging retired people inventing the future with makey makey. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3913–3922. ACM, 2014.

[159] Daniela K Rosner and Kimiko Ryokai. Reflections on craft: probing the creative process of everyday knitters. In *Proceedings of the seventh ACM conference on Creativity and cognition*, pages 195–204. ACM, 2009.

[160] Daniela K Rosner and Kimiko Ryokai. Spyn: augmenting the creative and communicative potential of craft. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2407–2416. ACM, 2010.

[161] Mark A Runco and Garrett J Jaeger. The standard definition of creativity. *Creativity research journal*, 24(1):92–96, 2012.

[162] Benjamin Samuel. *Crafting Stories Through Play.* PhD thesis, UC Santa Cruz, 2016.

[163] Sara. Eq 8 top 10 favorite features. https://doyoueq.com/blog/2017/11/eq8-top-10-1-enhanced-user-interface/. Accessed 21-September-2019.

[164] Mika Satomi. Nerding. http://www.nerding.at/. Accessed 21-September-2019.

[165] Owen Schaffer. Crafting fun user experiences: A method to facilitate flow. *Human Factors International*, 2013.

[166] Olive Schreiner. *From Man To Man Or Perhaps Only*. 1926.

[167] Kristin A Searle, Deborah A Fields, Debora A Lui, and Yasmin B Kafai. Diversifying high school students' views about computing with electronic textiles. In *Proceedings of the tenth annual conference on International computing education research*, pages 75–82. ACM, 2014.

[168] Kristin A Searle and Yasmin B Kafai. Boys' needlework: Understanding gendered and indigenous perspectives on computing and crafting with electronic textiles. In *ICER*, pages 31–39, 2015.

[169] Shima Seiki. Knitting machine function list: Needle. https://www.shimaseiki.com/product/knit/feature/. Accessed 21-September-2019.

[170] SHIMA SEIKI. Sds-one apex3. http://www.shimaseiki.com/product/design/sdsoneapex/flat/, 2011. Accessed 21-September-2019.

[171] Noor Shaker, Julian Togelius, and Mark J Nelson. *Procedural content generation in games*. Springer, 2016.

[172] Cathleen Shamieh. *Electronics For Kids For Dummies*. John Wiley & Sons, 2016.

[173] Sharon. Why crochet is unique (hint: Crochet machines do not exist). http://thesassyhook.com/blog/why-crochet-is-unique-hint-crochet-machines-do-not-exist/. Accessed 21-September-2019.

[174] Meir Shemla. Why workplace diversity is so important, and why it's so hard to achieve. https://www.forbes.com/sites/rsmdiscovery/2018/08/22/why-workplace-diversity-is-so-important-and-why-its-so-hard-\to-achieve/. Accessed 21-September-2019.

[175] Ben Shneiderman. Creativity support tools: Accelerating discovery and innovation. *Communications of the ACM*, 50(12):20–32, 2007.

[176] Ben Shneiderman, Gerhard Fischer, Mary Czerwinski, Mitch Resnick, Brad Myers, Linda Candy, Ernest Edmonds, Mike Eisenberg, Elisa Giaccardi, Tom Hewett, et al. Creativity support tools: Report from a us national science foundation sponsored workshop. *International Journal of Human-Computer Interaction*, 20(2):61–77, 2006.

[177] Jay Silver and Eric Rosenbaum. Makey makey. https://makeymakey.com/, 2012. Accessed 21-September-2019.

[178] Siren. Siren.care. https://siren.care/. Accessed 21-September-2019.

[179] SkyKnit. Skyknit: The collection. https://www.ravelry.com/patterns/library/skyknit-the-collection, 2018. Accessed 21-September-2019.

[180] Gillian Smith. Generative design for textiles: Opportunities and challenges for entertainment ai. In *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2017.

[181] Scott Sonenshein. How constraints force your brain to be more creative. https://www.fastcompany.com/3067925/how-constraints-force-your-brain-to-be-more-creative. Accessed 21-September-2019.

[182] Sparkfun. Lilypad products. https://www.sparkfun.com/lilypad_products. Accessed 21-September-2019.

[183] Sparkfun Electronics. Sparkfun electronics. https://www.sparkfun.com, 2003. Accessed 21-September-2019.

[184] Amy Spencer. *The crafter culture handbook*. Marion Boyars, 2007.

[185] Liza Stark. The soft circuiteer. http://thesoftcircuiteer.net/. Accessed 21-September-2019.

[186] George Stiny. Introduction to shape and shape grammars. *Environment and planning B: planning and design*, 7(3):343–351, 1980.

[187] George Stiny and James Gips. Shape grammars and the generative specification of painting and sculpture. In *IFIP Congress (2)*, volume 2, 1971.

[188] STOLL. M1plus pattern software. http://www.stoll.com/stollsoftwaresolutionsen4/patternsoftwarem1plus/31, 2011. Accessed 21-September-2019.

[189] Debbie Stoller. *Stitch'n bitch: The knitter's handbook*. Workman Publishing, 2012.

[190] Anne Sullivan and Emily K Johnson. Beaded adventures: Crafting stem learning. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 351–358. ACM, 2019.

[191] Anne Sullivan, Joshua Allen McCoy, Sarah Hendricks, and Brittany Williams. Loominary: crafting tangible artifacts from player narrative. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 443–450. ACM, 2018.

[192] Anne Sullivan, Anastasia Salter, and Gillian Smith. Games crafters play. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, page 26. ACM, 2018.

[193] Ivan Sutherland. *Sketchpad-a man-machine graphical interface*. PhD thesis, MIT, 1963.

[194] Joshua G Tanenbaum, Amanda M Williams, Audrey Desjardins, and Karen Tanenbaum. Democratizing technology: pleasure, utility and expressiveness in diy and maker practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2603–2612. ACM, 2013.

[195] Tap Happy. Royal tailor shop - prince & princess boutique (2.2.3977). `https://play.google.com/store/apps/details?id=com.kiwigo.royaltailorshopprinceprincess.free`, 2019. Mobile Application. Accessed: 2019-09-21.

[196] Klaus Teuber. Settlers of catan, 1995.

[197] Julian Togelius and Noor Shaker. The search-based approach. In *Procedural Content Generation in Games*, pages 17–30. Springer, 2016.

[198] Yarn Tree. Quarter stitches, half stitches. `https://www.yarntree.com/040halfstitch.htm`. Accessed 21-September-2019.

[199] Knit Picks Tutorials. Knitting in the round. `https://tutorials.knitpicks.com/knitting-in-the-round/`, 2009. Accessed 21-September-2019.

[200] UCSC. Games showcase 2016. `https://sammys.soe.ucsc.edu/2016/games`. Accessed 21-September-2019.

[201] Nobuyuki Umetani, Danny M Kaufman, Takeo Igarashi, and Eitan Grinspun. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph.*, 30(4):90, 2011.

[202] Jenny Underwood. The design of 3d shape knitted preforms. 2009.

[203] Urban Threads. Dark creatures - cat. `https://www.urbanthreads.com/products.aspx?productid=UT11804`, 2018. Accessed 21-September-2019.

[204] Arduino Users. Arduino project hub. `https://create.arduino.cc/projecthub/projects/tags/wearables`. Accessed 21-September-2019.

[205] Vincent van Gogh. Letter to theo van gogh. `http://www.webexhibits.org/vangogh/letter/14/378.htm`. Accessed 02-March-2017.

[206] Karen Vanderloock, Vero Vanden Abeele, Johan AK Suykens, and Luc Geurts. The skweezee system: enabling the design and the programming of squeeze interactions. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 521–530. ACM, 2013.

[207] Loïs Vanhée, Elena Márquez Segura, and Katherine Isbister. Firefly: A social wearable to support physical connection of larpers. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, page D311. ACM, 2018.

[208] Kiril Vidimče, Szu-Po Wang, Jonathan Ragan-Kelley, and Wojciech Matusik. Openfab: a programmable pipeline for multi-material fabrication. *ACM Transactions on Graphics (TOG)*, 32(4):136, 2013.

[209] Vogart. 12 honeymoon motifs for the romantic set (cats), 1950.

[210] Dhaval Vyas, Wim Poelman, Anton Nijholt, and Arnout De Bruijn. Smart material interfaces: a new form of physical interaction. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pages 1721–1726. ACM, 2012.

[211] Kai Wang and Jeffrey V Nickerson. A literature review on individual creativity support systems. *Computers in Human Behavior*, 74:139–151, 2017.

[212] Noah Wardrip-Fruin, Michael Mateas, Steven Dow, and Serdar Sali. Agency reconsidered. In *DiGRA Conference*, 2009.

[213] Philippa Warr. Threadsteading: A hex control game for the sewing machine. `https://www.rockpapershotgun.com/2016/10/17/threadsteading/`, 2016. Accessed 21-September-2019.

[214] Jo Waterhouse and Faythe Levine. *Indie craft*. Laurence King, 2010.

[215] Amy Wibowo, Daisuke Sakamoto, Jun Mitani, and Takeo Igarashi. Dressup: a 3d interface for clothing design with a physical mannequin. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, pages 99–102. ACM, 2012.

[216] Wikipedia contributors. Blackwork — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Blackwork`, 2008. [Accessed 02-March-2017].

[217] Therèsa M Winge and Marybeth C Stalp. Nothing says love like a skull and crossbones tea cozy: Crafting contemporary subversive handcrafts. *Craft Research*, 4(1):73–86, 2013.

[218] Peter Wonka, Michael Wimmer, François Sillion, and William Ribarsky. *Instant architecture*, volume 22. ACM, 2003.

[219] Cem Yuksel, Jonathan M Kaldor, Doug L James, and Steve Marschner. Stitch meshes for modeling knitted clothing with yarn-level detail. *ACM Transactions on Graphics (TOG)*, 31(4):37, 2012.

[220] Farnaz Zangouei, Mohammad Ali Babazadeh Gashti, Kristina Höök, Tim Tijs, Gert-Jan de Vries, and Joyce Westerink. How to stay in the emotional rollercoaster: lessons learnt from designing emroll. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, pages 571–580. ACM, 2010.